BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification : Not classified

A2

(11) International Publication Number: **WO 00/41533**

(43) International Publication Date: 20 July 2000 (20.07.00)

(21) International Application Number: PCT/US00/00688

(22) International Filing Date: 11 January 2000 (11.01.00)

(30) Priority Data:
09/228,249 11 January 1999 (11.01.99) US

(71) Applicant: INFOSPACE.COM, INC. [US/US]; 15375 N.E. 90th Avenue, Redmond, WA 98052 (US).

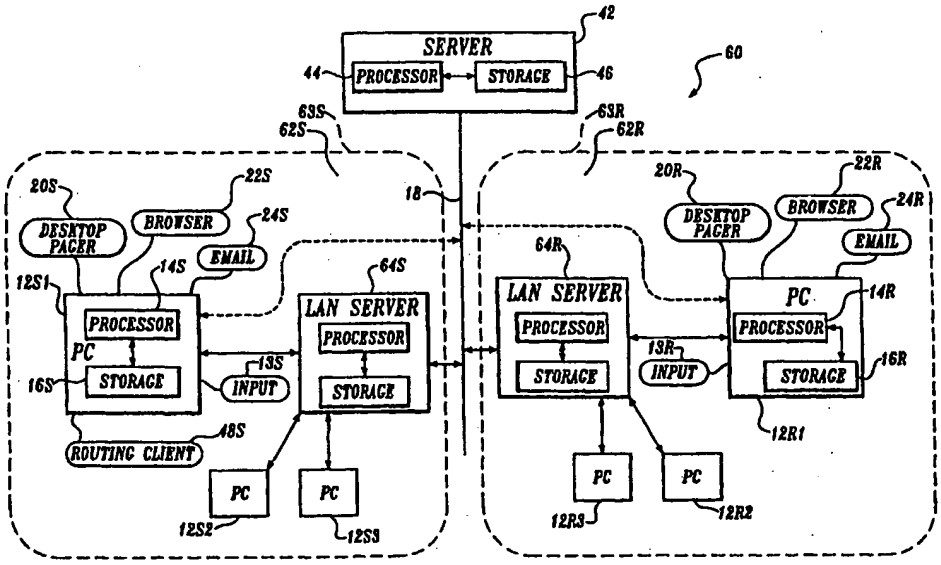
(72) Inventors: SHAH, Niraj, A.; 25744 S.E. 42nd Court, Issaquah, WA 98029 (US). HUGG, Ethan, B.; 6404 185th N.E., Seattle, WA 98155 (US). CHESTNUT, Kevin, L.; 210 12th Avenue East, Seattle, WA 98102 (US).

(74) Agent: TUMHEIM, Marcia, A.; Christensen O'Connor Johnson & Kindness PLLC, Suite 2800, 1420 Fifth Avenue, Seattle, WA 98101 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: IMPROVED SERVER AND METHOD FOR ROUTING MESSAGES TO ACHIEVE UNIFIED COMMUNICATIONS



(57) Abstract

A server routes one's incoming messages to his/her communication devices according to his/her routing preferences, and modifies the messages as needed. For example, the server can route an incoming page to one's email account and, if necessary, modify the page so that it is compatible with the email client. Thus, the server enables a number of diverse features such as: selection of routing topology (direct or indirect), translation of network restrictions, conditioning a synchronous communication for reception by an asynchronous device, message encryption, and callback or "buddy list" services.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

IMPROVED SERVER AND METHOD FOR ROUTING MESSAGES TO ACHIEVE UNIFIED COMMUNICATIONS

Technical Field:

The invention relates generally to communication networks that include computer hardware and software, and more particularly to a server, software run by the server, and a method implemented by the software for routing messages according to the message recipient's preferences.

Background of the Invention:

Today, a person may have more than one personal message device such as a wireless pager (*e.g.* a Skytel pager) or an e-mail client (*e.g.* Microsoft Outlook) that provides access to the person's e-mail account. Often, these devices communicate to other message devices via a computer network such as a local intranet or the Internet.

Figure 1 is a block diagram of a conventional computer network 10, which allows communication between message devices. The network 10 includes a sender's computer 12s, which has an input device 13s (*e.g.* a keyboard or a mouse) coupled thereto and which includes a processor 14s coupled to a storage device 16s. The network 10 also includes a recipient's computer 12r, which has an input device 13r and which includes a processor 14r and a storage device 16r. For example, the storage devices 16s and 16r may include a hard drive, volatile electronic memory, or both. The computers 12s and 12r are connected to a communication path 18 by networking circuitry that is omitted for clarity. For example, the path 18 may represent the communication lines that tie into and form the Internet. The processor 14s can run messaging devices such as a desktop pager 20s, a web browser 22s (*e.g.* Netscape Navigator), and an e-mail client 24s, which allows the sender to send and receive e-mail messages via an e-mail server 26s. Although the processor 14s executes the software that runs these devices, it is common to state that the computer 12s runs these devices. The sender may also have a wireless pager 28s and a voicemail server 30s, which are also connected to the path 18. The voicemail server 30s may allow the sender to send and receive voice messages via the computer 12s or via a telephone system (not shown).

Similarly, the recipient's computer 12r can run a desktop pager 20r, a web browser 22r, and an e-mail client 24r, which allows the recipient to view e-mail received on an e-mail server 26r. Also, the recipient may have a wireless pager 28r and a voicemail server 30r. Although the computers and message devices are labeled as sending or receiving devices for description purposes, it is understood that these labels are arbitrary such that the sending computer and message devices can be used to receive messages and the receiving computer and message devices can be used to send messages.

The system 10 may also include a file server 32, which is connected to the path 18 and which can assist with the transfer of messages between the sender's messaging devices and the recipient's messaging devices. For example, the server 32 may be a server of an internet service provider (ISP), which facilitates the transfer of messages between ISP account holders and between an account holder and a non-account holder. Or, the server 32 may be a paging company's server that transfers messages between the wireless pagers 28s and 28r.

In operation, the network 10 typically allows two topologies for transferring messages from one device to another: the point-to-point (PTP) topology, and the star topology. With the PTP topology, a message is routed directly between the sending and receiving devices. For example, using a PTP topology, the desktop pager 20s sends a message directly to the desktop pager 20r via the computer 12s, the path 18, and the computer 12r. In some applications, such as where it is an ISP server, the server 32 may open this direct path between the pagers 20s and 20r. Conversely, with a star topology, the message is routed through an intermediate node or device such as the server 32. For example, using a star topology, the pager 28s sends a message intended for the pager 28r to the server 32, which may be the paging company's server. The server 32 then processes the message and sends it to the pager 28r. This may occur for security or other reasons. Therefore, because the PTP topology eliminates the overhead of having the server receive and send the message, it is often faster and ties up fewer network resources than the star topology.

Unfortunately, if the environment of the network 10 does not allow all messages to be sent with a PTP topology, then the server 32 may be programmed to route all messages with a star topology to prevent messaging failure. This may

create an unnecessary bottleneck at the server 32, thus significantly increasing access times and aggravation for users of the server 32. Alternatively, if the same type of server 32 is to be installed in a network 10 having an environment that does allow all messages to be sent with a PTP topology, then the server software will have to be modified to allow this. Thus, if the server 32 can be used in both network environments, then the server manufacturer will have to develop and offer two respective software packages, one for PTP and another for star. Furthermore, the customer will have to install new software if the network environment changes, or if he wishes to install the server 32 in another network 10 having a different environment.

Furthermore, a recipient is often unable to retrieve messages from some of his message devices for extended periods of time, and if a message device is unavailable to receive a message, the message may be lost. For example, suppose the sender sends an e-mail message from his e-mail client 24s to the recipient's e-mail server 26r. If the recipient is out of town and has no access to the server 26r other than through the e-mail client 24r, then he must wait until he returns before he learns of and can read the sender's e-mail message. Alternatively, if the sender sends a desktop page from his pager 20s and the recipient's desktop pager 20r is not running, then the message has nowhere to go and may be lost.

Additionally, a message transfer may be unsuccessful if the sending device is of a different type than the receiving device. For example, if the recipient's e-mail client 24r is Microsoft Outlook, it may be unable to read an e-mail message from e-mail clients other than those sold by Microsoft.

Moreover, in applications where the server 32 is common to the sending and receiving devices, such as when it is an ISP server, the server 32 may use polling to allow a sender to determine if an intended recipient's message device is available to receive a message. For example, if the sender wants to send a desktop page, he may first want to determine if the intended recipient's computer is logged onto the server 32, and thus if the recipient is "online" and able to receive the page. To make this determination, the sender requests, via his computer 12s, the server 32 to poll all of the computers that are logged onto the server 32 and to notify the sender if one of these computer's is the recipient's computer 12r. Unfortunately, because the server 32 must communicate with each logged on computer, such polling requires a

significant amount of processing time, and thus can significantly increase user access times, particularly during hours of peak use. For example, it is common during peak hours for the number of logged-on computers to exceed one million! Furthermore, if the computer 12r is not logged onto the server 32 at the time that it performs the polling, then the only way for the sender to determine if the computer 12r subsequently logs on is to subsequently request the server 32 to repeat the polling. Thus, this significantly burdens the sender, because he may have to request several polls before he either gives up or the computer 12r logs onto the server 32.

SUMMARY OF THE INVENTION

In one aspect of the invention, a server is provided for facilitating communication between a sending device and a receiving device. The server includes a storage device for storing a program, and a processor for executing the program and having first and second states. The processor allows the sending device to send a message past the processor to the receiving device if the processor is in the first state, and the processor receives the message from the sending device and sends the message to the receiving device if the processor is in the second state.

Thus, such a server can automatically select and implement the best network routing topology, star or PTP, on a message-by-message basis. In one embodiment, the server selects and implements the PTP topology unless it cannot be implemented, in which case the server selects and implements the star topology.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a block diagram of a communications network according to the prior art.

Figure 2 is a block diagram of one embodiment of a communications network according to the invention.

Figure 3 is a block diagram of another embodiment of a communications network according to the invention.

Figure 4 is a flow chart of one embodiment of a procedure that the routing servers of Figures 2 and 3 implement to automatically set the network routing topology for transmission of a message.

Figure 5 is a computer screen generated by an embodiment of the message routing clients of Figures 2 and 3 for showing a message sender the available message devices of an intended message recipient.

Figure 6 is a web home page generated by an embodiment of the message routing server of Figures 2 and 3 for showing the available message devices of an account holder.

Figure 7 is a web page generated by an embodiment of the routing servers of Figures 2 and 3 for prompting a sender who is not logged onto the server for a message and other related information.

Figure 8 is a web page generated by an embodiment of the routing servers of Figures 2 and 3 for prompting a sender who is logged onto the server for a message and other related information.

Figure 9 is a flow chart of a message routing procedure that an embodiment of the routing servers and clients of Figures 2 and 3 implement.

Figure 10 is a computer screen generated by an embodiment of the routing clients of Figures 2 and 3 for prompting a recipient for his off-line routing preferences.

Figure 11 is a computer screen generated by an embodiment of the routing clients of Figures 2 and 3 for prompting a recipient for his on-line-but-unavailable routing preferences.

Figure 12 is a flow chart of a procedure implemented by an embodiment of the routing clients of Figures 2 and 3 for finding all of the message devices installed on the computers that respectively run the routing clients.

Figure 13 is a device-listing screen generated by the embodiment of the routing clients that implement the procedure of Figure 12.

Figure 14 is flow chart of a call-back procedure implemented by an embodiment of the servers and clients of Figures 2 and 3.

Figure 15 is a call-back-notification screen generated by the embodiment of the routing clients that implement the client portion of the call-back procedure of Figure 14.

Figure 16 is a flow chart of procedure implemented by an embodiment of the routing clients of Figures 2 and 3 for learning a recipient's messaging patterns and generating a routing preference based on these patterns.

Figure 17 is a redial screen generated by the embodiment of the routing clients that implement the procedure of Figure 16.

Figure 18 is a flow chart of a procedure implemented by one embodiment of the servers or clients of Figures 2 and 3 for setting client priority at log in if multiple clients of the same user are logged onto the server.

Figure 19 is a flow chart of a procedure implemented by one embodiment of the servers or clients of Figures 2 and 3 for setting client priority based on user activity if multiple clients of the same user are logged on to the server.

DETAILED DESCRIPTION OF THE INVENTION

Figure 2 is a block diagram of an embodiment of a communication network 40 according to the invention, where elements that are common to Figure 1 have the same reference numerals. The network 40 includes a routing server 42, which includes a conventional processor 44 and a conventional storage device 46. In one embodiment, the device 46 includes a volatile memory such as dynamic random access memory (DRAM), a non-volatile memory such as a hard disk, or a combination of both volatile and nonvolatile memory. The processor 14r of the computer 12r runs a routing client 48r, which, as discussed below, works with the server 42 to route the recipient's messages according to the recipient's message routing preferences. The processor 14s of the sender's computer 12s may also run a routing client 48s, which in one embodiment is the same as the routing client 48r. In one embodiment, the server 42 runs My Agent server software from Active Voice Corporation, and the clients 48s and 48r are My Agent software clients from Active Voice.

Still referring to Figure 2, and as discussed in more detail below in conjunction with Figures 4-19, the general operation of the network 40 is discussed according to one embodiment of the invention.

In operation, the server 42 routes the recipient's incoming messages to the recipient's message device specified by the recipient's routing preferences. For example, the routing preferences may specify that the server 42 route all messages directed to the desktop pager 20r to the e-mail server 26r.

To allow the server 42 to perform such rerouting, the recipient gives the sender access to one or more of the recipient's message devices via the server 42.

In one embodiment, this access is through the sender's routing client 48s, the recipient's web page set up on the server 42, or the recipient's address with respect to the server 42.

The server 42 automatically determines the best network topology for routing a message from the sending device to the receiving device specified by the recipient's routing rules based on criteria including the sender's identity, the identity of the recipient's message device to which the sender has directed the message, the priority of the message (*e.g.,* urgent, normal, or low), the receiver's availability, and the size of the message. In one embodiment, the server 42 routes the message using a PTP topology unless this topology is unavailable with respect to the message.

In one embodiment, if the format, such as the protocol, size, or encryption, of the sent message is incompatible with the receiving device specified by the recipient's routing preferences, then the server 42 reformats the message before sending it to the receiving device. Thus, the server 42 allows one type of message device, such as the web browser 22s, to send a message to another type of message device, such as a desktop pager 20r.

In another embodiment, the server 42 eliminates the problems with conventional polling by maintaining a list of the users that are currently logged onto the server 42. This allows a user to request a "callback" from the server 42 when another user logs onto the server 42.

In yet another embodiment, the client 48r monitors the recipient's patterns with respect to his received messages, and based on these patterns, automatically suggests, develops, or maintains the routing preferences that best fit the recipient's lifestyle.

In still another embodiment, the server 42 allows a user to have multiple computers 12r simultaneously logged onto the server 42, where each computer 12r is running a respective routing client 48r. For example, it is common for a user to have a work computer and a home computer. Thus, the server 42 allows both of these computers to be simultaneously logged on and running respective routing clients 48r. To prevent conflicts if the clients 48r have different routing preferences, the clients 48r determine which of them is the primary client whose routing rules the server 42 will follow.

Figure 3 is a block diagram of a communications network 60 according to another embodiment of the invention, where like elements have like reference numerals with respect to Figures 1 and 2. In the network 60, the computers 12s1 and 12r1 are part of local area networks 62s and 62r, respectively. Each of the networks 62s and 62r is protected by a respective conventional firewall, represented by the dashed lines 63s and 63r, respectively, and includes a respective server 64s and 64r. In one embodiment, the communication path 18 represents the Internet, the computer 12s and the server 64s communicate with each other over an intranet, and the computer 12r and the server 64r communicate with each other over another intranet. Furthermore, each of the networks 62s and 62r is similar to the network 40 of Figure 2, where the servers 64s and 64r each correspond to the server 42 of Figure 2. Thus, in this embodiment, the server 64s routes messages between the message devices of the network 62s in a manner similar to that described for the server 42 of Figure 2. Likewise, the server 64r routes messages between the message devices of the network 63r in a similar manner.

Still referring to Figure 3, despite the firewalls 63s and 63r, the server 42 allows a sending device in the network 62s to send a message to a receiving device in the network 62r and routes the message according to the recipient's routing rules. Typically, the firewalls 63s and 63r prevent the server 42 from implementing a PTP topology for such a message. But because the server 42 can automatically select the proper topology, the same server 42 that is used in the network 40 of Figure 2 can also be used in the network 60. That is, neither the server hardware nor server software need be modified, so manufacturing and installation expenses are reduced compared to prior-art communication servers.

Figure 4 is a flow chart that details one embodiment of the general topology selection and message routing procedure used by the networks 40 and 60 of Figures 2 and 3, respectively. For clarity, reference will be made to the elements of Figure 2 unless otherwise specified.

Referring to step 70, the sending device, for example the desktop pager 20s, initiates the sending of a message to a receiving device by sending a conventional message-initiation header to and requesting the IP address and dynamic encryption key of the receiving device (or of the computer, such as the computer 12s, running the device) from the routing server 42 via the path 18. With respect to the network

60 of Figure 3, however, the pager 20s typically sends this information to the path 18 via the server 64s. The message-initiation header typically includes information such as the identities of the sender and recipient and the length and priority of the message. Furthermore, in one embodiment, the server 42 determines the identities of the sending and intended receiving devices from the format of the message header. For example, a header from the desktop pager 20s often has a different number of bytes or is otherwise different than a header from the web browser 22s.

Next, referring to steps 72 and 73, the server 42 examines the message-initiation header and, based on the header, the network environment, and the recipient's routing rules, determines the appropriate receiving device and whether or not PTP communication between the sending and receiving devices is possible.

For example, suppose the sender desires to send a message from his desktop pager 20s to the recipient's desktop pager 20r. Furthermore, suppose that the recipient's routing rules indicate that the desktop pager 20r is to receive this message. If the server 42 determines that there are no firewalls or other network environment conditions that prevent a PTP topology, it implements a PTP topology.

Alternatively, suppose the sender desires to send a message from his e-mail client 24s to the recipient's e-mail account on the e-mail server 26r, and that the recipient's routing rules instruct the server 42 to route all messages directed to the e-mail server 26r to the desktop pager 20r. If the format of the message from the e-mail client 24s in incompatible with the desktop pager 20r, then the server 42 determines that a star topology is appropriate so that the server 42 can receive and reformat the message from the e-mail client 24s and then send the reformatted message to the desktop pager 20r. For example, desktop pagers such as the desktop pager 20r often limit the size of a received message to 100 – 200 bytes. Therefore, if the message from the e-mail client 24s is longer than this, the server 42 will decide on a star topology so that it can receive and truncate the message before sending it to the desktop pager 20r.

Or, if the message is so large or has so many recipients that a PTP topology would be unable to efficiently handle the message, the server 42 may implement the star topology. For example, suppose the sender wishes to send an e-mail message having a one-megabyte attachment to ten recipients, and that all of the recipients' routing rules indicate that the server 42 is to route such an e-mail message to their

respective e-mail servers 26r. In one embodiment, because of the file length and the relatively large number of recipients, the server 42 determines that multicasting is more efficient than setting up direct PTP paths between the sender's e-mail server 26s and the respective e-mail servers 26r. Therefore, the server 42 implements a star topology by instructing the e-mail server 26s to send the message to the server 42 only once, and then sending the received message to each of the e-mail servers 26r of the respective recipients. Alternatively, the server 42 may forward the message to a conventional multicasting server (not shown), which sends the message to each of the e-mail servers 26r.

Moreover, the server 42 may allow the sending device, such as the desktop pager 20s, to first try to send a message with a PTP topology, and if this attempt fails, the server 42 instructs the sending device to retry with a star topology.

Referring to Figure 3, the server 42 may implement variations of the star topology in the network 60 if one or both of the firewalls 63s and 63r prevent the server 42 from opening a PTP path between a message device of the network 62s and a message device of the network 62r. In one embodiment, after determining that it cannot implement a PTP topology, the server 42 first tries to implement a version of the star topology in which the server 42 bypasses the servers 64s and 64r and communicates directly with the sending and receiving devices. This is significantly faster and causes less traffic on the networks 62s and 62r than if the message were routed through the servers 64s and 64r. For example, if the desktop pagers 20s and 20r are the sending and receiving devices respectively, then the server 42 receives the message from the pager 20s and sends it to the pager 20r in a manner similar to that described above with respect to a star topology in the network 40 of Figure 2. If the server 42 cannot implement this version of the star topology, then, as a last resort, the server 42 routes the message through one or both of the servers 64s and 64r.

Next, referring to step 75, if a PTP topology is possible, then the server 42 sends the IP address and the dynamic encryption key of the receiving device specified by the routing preferences (or of the computer 12r if it is running the receiving device) to the sending device.

Then, referring to step 77, the sending device sends the message directly to the receiving device — thus bypassing the server 42, and with respect to the network

60 of Figure 3, bypassing the servers 64s and 64r — and, after it sends the message, conventionally closes the direct PTP communication path over which the sending device sent the message.

Alternatively, referring to step 79, if the server 42 cannot implement a PTP topology, the server 42 implements a star topology. Specifically, the server 42 opens a communication path between itself and the sending device and notifies the receiving device specified by the recipient's routing rules of the incoming data steam that forms the message. For example, as discussed above, if the e-mail client 24s is the sending device and the desktop pager 20r is the receiving device, then the server 42 opens a path between the e-mail client 24s and itself via the e-mail server 26s, and notifies the desktop pager 20r that a message is forthcoming.

Next, referring to step 81, the sending device transfers the message to the server 42.

Then, referring to step 83, the server 42 reformats the message if necessary and then sends the message to the specified receiving device. For example, if the e-mail client 24s is the sending device and uses a first message format and desktop pager 20r is the receiving device and uses a second message format, the server 42 converts the message from the e-mail client 24s into the second format, and then transfers the reformatted message to the desktop pager 20r.

Next, referring to step 85, when the sending device finishes sending the message, it notifies the routing server 42, which conventionally closes the communication path between itself and the sending device.

Then, referring to step 87, the server 42 conventionally closes the communication path between itself and the receiving device.

Thus, the servers 42 of the networks 40 and 60 of Figures 2 and 3, respectively, can facilitate more efficient communication between message-sending and message-receiving devices by automatically selecting the best network communication topology. Also, the servers 42 allow a recipient to redirect a message from one receiving device to another receiving device, and allow a message device of one type to communicate with a message device of another type.

Figures 5 - 8 disclose embodiments of techniques that allow a sender to send a message to the recipient such that the server 42 can route the message according to the recipient's routing preferences. Figures 5 – 8 are discussed in conjunction

with the network 40 of Figure 2, it being understood that the discussion is also applicable to the network 60 of Figure 3 unless otherwise noted.

Figure 5 is a computer screen 90 that allows a sender who is a registered user of the routing server 42 to send messages to a recipient who is also a registered user of the server 42. Using the routing client 48s, the sender creates one or more groups of recipients, and adds the recipient to one of these groups. For example, a sender may have a group for work colleagues and another group for personal friends. The client 48r for each designated recipient prompts the respective recipient for messaging information, receives the information from the recipient, and makes this information available to the sender via the server 42. Based on this information, the routing client 48s generates the screen 90 on the sender's computer 12s.

The screen 90 includes a list field 92, which includes a list of messaging devices that the recipient has made available to receive messages from the sender. In one embodiment, the routing client 48s is run in a Microsoft Windows® environment so that the sender can select the desired messaging device by pointing and clicking with a mouse. For example, if the sender points and clicks on the "Page" icon, then the routing client 48s will prompt the sender to enter a message to the desktop pager 20s, which will send the message to the recipient's desktop pager 20r (or other message device specified by the recipient's routing rules) with the help of the server 42 as discussed above in conjunction with Figure 4. In one embodiment, some messaging devices such as the desktop pager 20s and a chat device (activated by clicking on the "Chat" icon) actually run as part of the routing client 48s. But the routing client 48s operates in a similar manner for other message devices as well. For example, the field 92 allows the sender to send messages to the recipient's e-mail server 26r, fax, or telephone. In response to the sender's selection of these devices, the routing client 48s respectively activates the sender's e-mail client 24s or modem (not shown) so that the sender can proceed to send the message to the respective receiving devices. Furthermore, although icons are shown for certain messaging devices, the field 92 may include icons for other messaging devices such as but not limited to a wireless pager (*e.g.* Skytel®) or a personal digital assistant (PDA).

Other features of the screen 90 include an image field 98, which can include the recipient's photo or a live picture, a greeting field 100, which can include the recipient's greeting, and a log-in status field 102, which indicates whether the recipient — or more accurately the computer 12r running the client 48r — is logged onto the server 42. The screen 90 may also include other fields such as a schedule field that includes the recipient's current calendar.

Figures 6 and 7 are web pages that allow a sender who is not registered user of the routing server 42 to send messages via the web browser 22s to a recipient who is a registered user of the server 42.

Figure 6 is a recipient's home page 104 on the server 42. The sender accesses the home page 104 by using his web browser 22s to access the URL for the home page 104. Like the screen 90 of Figure 5, the page 104 includes a device field 106, a greeting field 108, a log-in status field 110, and an image field 114, and may include other fields such as a schedule field. Like the screen 90, although icons for certain messaging devices are shown, the device field 106 may include icons for other messaging devices such as but not limited to a wireless pager (*e.g.* Skytel®) or a personal digital assistant (PDA).

The sender uses the web browser 22s to send a message to a receiving device selected from the field 106, and as discussed above in conjunction with Figure 4, the server 42 reformats the message if necessary and routes the message to the receiving device specified by the recipient's routing preference. In one embodiment, the page 104 also includes an option field 116. The "My Groups" option allows the sender to view the groups to which the recipient belongs. The "My Profile" option allows the sender to view the recipient's profile, which includes additional information about the recipient. The "Search My Agent" option allows the sender to access the web pages of other registered users of the server 42 without knowing their URLs. This option is also available from the general home page (not shown) of the server 42. A user, however, may instruct the server 42 to prohibit others from accessing his web page through the "Search My Agent" option for security or privacy reasons.

Figure 7 is a page 120, when the server 42 sends the web browser 22s if the sender clicks on the "My Email" icon on the page 104 of Figure 6. The screen 120 prompts the sender for information and allows the sender to send an e-mail message

to the recipient via the web browser 22s. As discussed above in conjunction with Figure 4, the server 42 routes this e-mail message to the recipient's e-mail server 26s or to another of the recipient's message devices according to the recipient's routing preferences.

Figure 8 is a screen 122, which allows a registered user of the server 42 to send a message from the user's own web site to a registered or unregistered recipient. The screen 122 prompts the sender for the necessary information, such as the recipient's user name or e-email address. The screen 122 also includes a "Group Options" field, which allows the user to form and join user groups, to invite other registered users to join a group, and to unjoin groups.

Referring to Figures 9 through 11, embodiments of the techniques for setting a recipient's routing preferences and routing messages according these routing preferences are discussed.

Figure 9 is a flow chart showing how the server 42 and the receiving client 48r route messages according to an embodiment of the invention. The flow chart of Figure 9 is similar to the flow chart of Figure 4, except that it focuses on message routing instead of on the determination of the network topology.

Referring to step 130, the server 42 receives the message-initiation leader from the sending device. Next, referring to step 132, the server 42 determines whether or not the recipient's computer 12r, which runs the client 48r, is logged onto the server. If not, the server 42 routes the message according to the recipient's off-line routing preferences. For example, in one embodiment, if the recipient's device to which the sender directed the message is unavailable, then referring to step 134, the server 42 notifies the sender that the intended receiving device is unavailable. The server 42 may give the sender the option of sending the message to the receiving device specified by the off-line routing preferences or of canceling the message. Next, referring to step 136, if the sender elects to send the message, then the server 42 routes the message to the receiving device specified by the recipient's off-line routing preferences. For example, suppose that the sender wants to send a message from the desktop pager 20s to the desktop pager 20r but the computer 12r is not logged onto the server 42 via the client 48r. Furthermore, suppose that the recipient's routing preferences instruct the server 42 to route desktop pages to the e-mail server 26r if the computer 12r is off line. Thus, the server 42 informs the sender

that any page he sends will be routed to the recipient's e-mail server 26r and asks the sender if he still wants to send the page or if he wants to cancel and wait until later. If the sender decides to go ahead and send the page, the server 42 will route the page to the email server 26r. In another embodiment, however, the server 42 routes the message to the preferred off-line device without informing the sender.

Referring to step 138, if the computer 12r is logged onto the server 42, then the server 42 routes the message to the receiving device specified by the recipient's online routing preferences. For example, the on-line routing preferences may instruct the server 42 to route a page from the desktop pager 20s to the desktop pager 20r.

Next, referring to step 140, after the server 42 routes the message, the receiving client 48r determines if the specified receiving device has a rerouting condition, such as a user-activity rerouting condition, associated with it. If there is no condition, then referring to step 142, the server 42 and the client 48r take no further action with respect to the message. If there is a rerouting condition, however, then referring to step 144, the client determines if the condition is met. If the condition is met, then referring to step 146, the client causes the server to reroute the message to the device specified by the routing preferences. For example, as discussed below in conjunction with Figure 11, the routing preferences may specify that if a recipient does not "pick up" a message to the desktop pager 20r within a certain amount of time, then the client 48r is to cause the server 42 to reroute the message to another receiving device such as the e-mail server 26r. Thus, if the recipient does not pick up the page within the allotted time, then the client 48r causes the server 42 to reroute the page to the e-mail server 26r. Referring again to steps 144 and 146, in one embodiment, the client 48r monitors the receiving device to determine if the condition is met. This embodiment is useful when the receiving device, for example the desktop pager 20r, is part of the client 48r. In another embodiment, the receiving device notifies the client when the condition has been met.

Figure 10 is a screen 147, which is generated by the routing client 48r and which prompts a recipient to enter his off-line routing preferences. Specifically, a prompt 148 prompts the recipient to select the preferred device or devices for receiving a message intended for the desktop pager 20r if the computer 12r is not logged onto the server 42 when the message is sent. In the embodiment shown, the

recipient enters the preferred device or devices, here the e-mail server 26r, in a field 149. Thus, if the recipient is out of town and is not running his computer 12r, then the server 42 will forward all desktop pages to his e-mail server 26r. If the recipient has remote access to his e-mail server 26r, then he can access these desktop pages before he returns from his trip.

Figure 11 is a screen 150, which is generated by the routing client 48r and which prompts the recipient to enter a rerouting condition. Specifically, a prompt 151 prompts the recipient to check a box 152 if he would like the server 42 to reroute desktop pages if the recipient does not pick up the message within a time period specified in a box 154. The device to which the page will be rerouted is specified in a box 156.

The server 42 or the client 48r can determine if the recipient has picked up the desktop page from the desktop pager 20r in a number of ways. In one embodiment, the client 48r or the server 42 monitors the input device 13r to determine if it has provided any data to the computer 12r within the time period specified in the box 154. The idea is that if the input device 13r provides data during the specified time period, then the recipient was sitting at the computer 12r during this period and thus has read the desktop page. Conversely, if the input device 13r has not provided data, then the recipient was not sitting at the computer 12r during the specified period and thus has not read the desktop page. The input device 13r may be any conventional input device such as a keyboard or mouse. Alternatively, the device 13r may be a device such as a video camera or a microphone that the server 42 or client 48r monitors for movement or sound, respectively.

Figure 12 is a flow chart of an automatic-message-device-recognition procedure implemented by one embodiment of the routing client 48r.

First, referring to the step 160, the recipient boots the routing client 48. The recipient may do this by a special command after the computer 12r has booted up, or the client 48r may boot automatically during the boot sequence of the computer 12r.

Next, referring to step 162, the booted client 48r searches the storage area 16r of the computer 12r for message devices that are installed on the computer 12r such as the desktop pager 20r, the web browser 22s, and the e-mail viewer 24s.

Then, referring to step 164, the routing client 48r determines which of these installed message devices the recipient wants to make available to senders. In one

embodiment, these available message devices are included in the device fields 92 and 106 as discussed above in conjunction with Figures 5 and 6, respectively. More specifically, on its first boot, the client 48r includes all such devices in the fields 92 and 106. The recipient, however, can remove one or more of these devices from the fields 92 and 106. On subsequent boots, the client 48r will omit from the fields 92 and 106 any message devices previously removed therefrom, unless the recipient subsequently adds these devices back to the fields 92 and 106.

Next, referring to the step 166, the booted client 48 sends to the server 42 the identifies, addresses, and other information for the message devices that are listed in the fields 92 and 106, and also sends the server 42 the recipient's routing preferences as discussed above.

Therefore, the recipient does not have to perform a tedious installation of the message devices into the client 48r or the server 42. Furthermore, the client 48r may even identify and list message devices that the recipient didn't even know were installed on the computer 12r!

Figure 13 is a display screen 170, which one embodiment of the client 48r generates according to the flow chart of Figure 12 to allow a recipient to remove and add message devices that are available to senders. The available devices are listed in a field 172, and can be deleted or added by clicking on the "Delete Device" and "Add Device" icons, respectively. When the "Add Device" icon is selected, the client 48r lists for the recipient's selection all message devices installed on the computer 12r but not currently available to senders, *i.e.*, not listed in the fields 92 or 106.

Figure 14 is a flow chart of a callback procedure executed by the server 42 and the routing client 48s according to an embodiment of the invention.

Referring to step 180, the server 42 maintains a list of the users that are currently logged onto the server 42 via their respective clients 48, and also maintains a list of undelivered callback requests.

Next, referring to steps 182, 184 and 186, in one embodiment, the server 42 provides to a sender the log-on status of the recipients in the sender's groups, such as provided in the field 102 of the screen 90 in Figure 5. More specifically, referring to step 182, the sender logs onto the server 42 via the computer 12s and the client 48s. Next, referring to step 184, the server 42 determines the log-on status of each user in the sender's groups by checking the logged-on list. In one embodiment, the

server 42 stores the membership data for the sender's groups so that the client 48s need not send this data to the server every time the sender logs onto the server. Then, referring to step 186, the server 42 sends the log-on status of each of these users to the sending client 48s. In one embodiment, the sending client 48s can also request the log-on status of a user even after the sending client 48s has logged onto the server 42.

Next, referring to step 188, the sender requests a callback. That is, the sender requests the server 42 to deliver a callback request to the client 48r of a recipient. The callback request notifies the recipient that the sender wishes to contact him/her. For example, in one embodiment, referring to the field 92 in the screen 90 of Figure 5, the sender can request a callback by clicking on the "Set A Callback" icon.

Then, referring to steps 190 and 192, the server 42 checks the logged-on list and determines whether the recipient is logged onto the server.

Next, referring to step 194, if the recipient is logged on, then the server delivers the callback request to the recipient's client 48r.

But, referring to step 196, if the recipient is not logged on, then the server adds the callback request to the callback list. Referring to step 198, when the recipient logs on, the server 42 checks the callback list to determine if the recipient has any outstanding callback requests. If, as in this example, the recipient does have an outstanding callback request, then the server 42 delivers the callback request to the recipient's client 48r.

Thus, the callback procedure eliminates the problems associated with conventional polling as discussed above in conjunction with Figure 1.

Referring to Figure 15, in one embodiment of the callback procedure described in the flow chart of Figure 14, the client 48r generates a screen 200 in response to the callback request delivered by the server 42. The screen 200 identifies the sender and allows the recipient, via the client 48r, to either allow or cancel the callback. That is, the recipient has the option of allowing the server 42 to notify the sender that the recipient is now available or of preventing the server 42 from doing so. Thus, the recipient can cancel the callback request if he/she does not want to be bothered by the sender.

Figure 16 is a flow chart of a message-routing learning procedure implemented by one embodiment of the routing client 48r. Implementing this procedure allows the client 48r to automatically suggest, generate, or maintain the recipient's routing preferences.

Referring to step 201, the client 48r periodically or continually monitors the recipient's actions with respect to his received messages. Next, referring to step 202, the client 48r automatically suggests changes to, sets, or updates the routing preferences based on patterns that the client 48r has detected with respect to the received messages and the recipient's related actions. Then, referring to step 204, the client 48r sends these new routing preferences to the server 42 (with the recipient's permission if the client 48r has only suggested new routing preferences).

Still referring to steps 201, 202, and 204, in one embodiment, the client 48r implements a statistical correlation matrix, such as a conventional Baysian network, to correlate message characteristics (e.g., sender's identity, time of day message received) with the recipient's actions (e.g., forward or ignore message) for a group of messages such as the last one thousand received messages.

For example, suppose that using this technique, the client 48r determines that out of fifty phone calls to the recipient's work phone on weekends and after 5:00 p.m. on weekdays, the recipient has answered two, and the rest have been routed to the recipient's voicemail server 30r. (In one embodiment, the client 48r can determine whether a call is answered or sent to voice mail by communicating with the voicemail server 30r using conventional techniques.) Therefore, in response to this pattern, the client 48r may suggest that the recipient adopt a routing preference that causes the server 42 to route all work phone calls received on weekends and after 5:00 p.m. and on weekdays directly to the voicemail server 30r, and thus reduce the chances that the caller will be aggravated by the phone ringing a number of times before he is transferred to voicemail.

Or, suppose that the client 48r determines that out of twenty five e-mail messages from a particular sender when the e-mail client 24r also displays unread e-mail messages from other senders, the recipient has opened this particular sender's messages first twenty four times. (In one embodiment, the client 48r can determine the order in which unread e-mail messages are opened by communicating with the e-mail client 24r or e-mail server 26r using conventional techniques.) In

response to this pattern, the client 48r may suggest that the recipient adopt a routing preference that causes the server 42 to route all e-mails from this particular sender with high priority or in another manner such that they are always at the top of the unread e-mail list when the e-mail client 24r displays unread e-mail messages.

Figure 17 is a screen 206 and a redial list 208 generated by one embodiment of the routing client 48s according to a procedure similar to that discussed above in conjunction with Figure 16. Unlike the Figure 16 procedure, however, this procedure learns a senders message-sending patterns. More specifically, the client 48s keeps track of the most popular message-sending actions that the sender has taken. In this embodiment, the client 48s retains ten actions, and updates the list 208 to include the last action taken. But when the client 48s updates the list 208 with the most recent action, it removes the least popular action on the list 208 and not necessarily the least recent action taken. Thus, the list 208 is not merely a list of the last ten actions taken, but is a combination of the last actions taken and the actions that the sender takes most frequently. For example, the list 208 may include the last five actions taken, and five of the most frequently taken actions.

Figures 18 and 19 are flow charts showing embodiments of respective procedures that allow a user to have multiple routing clients 48 simultaneously logged onto the server 42. For example purposes, referring to Figure 2, assume that the recipient owns the computers 12s (work) and 12r (home) and runs the routing clients 48s and 48r simultaneously. As discussed above, the labels of sending and receiving for the clients 48s and 48r are arbitrary as these clients can perform both message-sending and message-receiving functions. Therefore, this is an accurate example.

The flow chart of Figure 18 is an embodiment of a procedure to designate a newly logged-on client 48 as the primary client and the other client or clients that are already logged on as passive clients. The significance of the primary client 48 is that the server 42 follows the routing preferences of the primary client. For example purposes, the client 48s is the newly logged-on client, and the client 48r is already logged on to the server 42 at the time the client 48s logs on. It is understood, however, that in some embodiments there may be more than one client 48 already logged onto the server 42.

More specifically, referring to step 220, the "new" client 48s logs onto the server 42 via the computer 12s and determines whether or not the client 48r is logged onto the server 42. The new client 48s may make this determination in a variety of ways. In one embodiment, the server 42 automatically provides the new client 48s with the log-in status of the client 48r in a manner similar to that discussed above in conjunction with Figure 14. In another embodiment, the new client 48s requests the log-in status of the client 48r from the server 42 also in a manner similar to that discussed above in conjunction with Figure 14.

Next, referring to step 222, if the client 48r is not logged onto the server 42, then, referring to step 224, the new client 48s transfers its message-routing preferences to the server 42, and referring to step 226, the server 42 proceeds to route messages according to these routing preferences as discussed above in conjunction with Figure 4.

On the other hand, if the client 48r is logged onto the server, then the client 48s instructs the client 48r to become passive. The client 48s may provide these instructions to the client 48r in a number of ways. In one embodiment, the new client 48s requests the server 42 to set up a PTP communication path between it and the client 48r as discussed above in conjunction with Figure 4. In other embodiments, the new client 48r requests a communication path to the client 48r through the server 42 (star topology) also as discussed above in conjunction with Figure 4, or the server 42 instructs the client 48r to become passive.

Referring again to steps 224 and 226, after the client 48r is instructed to become passive, then the new client 48s transfers its routing preferences to the server 42, which routes messages according to these preferences.

The flow chart of Figure 19 shows an embodiment of a procedure to select a new primary client among multiple clients that are all already logged onto the server 42.

Referring to step 230, multiple clients 48 are logged onto the server 42, and one of these clients is the primary client and the others are passive clients. For example purposes, suppose that the user went home from work and left his work client 48s running. Then suppose he logs the home client 48r onto the server 42, and according to the procedure described in conjunction with Figure 18, the client 48r becomes the primary client and the client 48s becomes the passive client.

Referring to step 232 and using the above example, the passive client 48s detects a condition, such as user activity, that indicates it should now be the primary client. For example, this situation occurs if the user goes back to work without logging off the client 48r and starts using the computer 12s. The theory here is that the user wants the client on the computer he is using, here the client 48s, to be the primary client so that his messages are routed accordingly. In one embodiment, the client 48s detects the user activity by monitoring the input device 13s as discussed above in conjunction with Figure 9.

Next, referring to step 234, the passive client 48s instructs the primary client 48r to become passive. In one embodiment, the passive client 48s communicates with the client 48r as discussed above in conjunction with Figure 18.

Then, referring to the step 236, the passive client 48s transfers its message routing preferences and other information to the server 42 and becomes the new primary client.

Referring to step 238, the server 42 then routes subsequent incoming messages according to the routing preferences provided by the new primary client 48s.

From the foregoing it will be appreciated that, although specific embodiments of the invention have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention.

What is claimed:

1. A server for facilitating communication between a sending device and a receiving device, the server comprising:

a storage device operable to store a program; and

a processor coupled to the storage device, operable to execute the program, and having first and second states, the processor operable to allow the sending device to send a message past the processor to the receiving device if the processor is in the first state, the processor operable to receive the message from the sending device and to send the message to the receiving device if the processor is in the second state.

2. The server of claim 1 wherein the processor is operable to receive a message-initiation header from the sending device and to enter one of the first and second states in response to the message-initiation header.

3. The server of claim 1 wherein the processor enters the second state only if the sending device cannot send the message directly to the receiving client.

4. The server of claim 1 wherein the processor is operable to receive a message routing preference of a user of the receiving device and to enter one of the first and second states in response to the message routing preference.

5. A server for allowing communication between a sending device of a first type and a receiving device of a second type, the server comprising:

a storage device operable to store a program; and

a processor coupled to the storage device and operable to execute the program, receive a message routing preference of a user of the receiving device, and to route a message from the sending device to the receiving device in response to the message routing preference.

6. The server of claim 5 wherein the processor is operable to receive the user's message routing preference from a client of the user when the client is logged on to the server.

7. The server of claim 5 wherein the processor is operable to route the message to the receiving device upon occurrence of a condition specified by the message routing preference.

8. The server of claim 5 wherein the processor is operable, in response to the message routing preference, to route the message to the receiving device when no client of the user is logged onto the server.

9. The server of claim 5 wherein the processor is operable to receive the message from the sending device, to convert the message into a form that is compatible with the receiving device, and to send the converted message to the receiving device.

10. The server of claim 5 wherein the processor is operable to route the message to the receiving device upon the occurrence of a condition specified by the message routing preference.

11. The server of claim 5 wherein the processor is operable to first allow the sending device to send the message to a receiving device of the first type and then to route the message to the receiving device of the second type upon the occurrence of a condition specified by the message routing preference.

12. A server for allowing communication between a first receiving device of a user and a sending device, the server comprising:

a storage device operable to store a program; and

a processor coupled to the storage device, operable to execute the program, operable to receive a message routing preference of the user, and, in response to the message routing preference, operable to route to the first receiving device a message that the sending device directed to a second receiving device of the user.

13. The server of claim 12 wherein the processor is operable to route the message to the first receiving device upon occurrence of a condition specified by the message routing preference.

14. The server of claim 12 wherein:

the second receiving device comprises a desktop pager of the user; and

the processor is operable, in response to the message routing preference, to route the message to the first receiving device if no computer logged onto the server is running the user's desktop pager.

15. The server of claim 12 wherein:

the message is sent by the sending device in a form that is compatible with the second receiving device; and

the processor is operable to receive the message from the sending device, to convert the message into a form that is compatible with the first receiving device; and to send the converted message to the first receiving device.

16. The server of claim 12 wherein:

the second device comprises a desktop pager; and

the processor is operable to route the message to the first receiving device after a period of user inactivity on a computer that is running the desktop pager, the message routing preference specifying the duration of the period.

17. The server of claim 12 wherein:

the second device comprises a desktop pager; and

the processor is operable to allow the sending device to send the message to the desktop pager, to retrieve the message from the desktop pager, and route the message to the first receiving device after a period of user inactivity on a computer that is running the desktop pager, the message routing preference specifying the duration of the period.

18. A server for communicating with a client, the server comprising:

a first storage device operable to store a program; and

a processor coupled to the storage device and operable to execute the program and to store a log-on status of the client.

19. The server of claim 18 wherein the processor is operable to store the log-on status in the first storage device.

20. The server of claim 18 wherein:

the processor comprises a memory; and

the processor is operable to store the log-on status in the memory.

21. The server of claim 18, further comprising:

a second storage device; and

wherein the processor is operable to store the log-on status in the second storage device.

22. The server of claim 18, further comprising:

a log-on-status storage device;

wherein the processor is operable to, further comprising:

maintain a record of logged-on clients in the a log-on-status storage device; and

wherein the processor is operable to store a value in the record while the client is logged onto the server, the value identifying the client.

23. A server for first and second clients, the server comprising:

a storage device operable to store a program; and

a processor coupled to the storage device and operable to execute the program, to receive a callback request from the first client, and to provide the callback request to the second client.

24. The server of claim 23, further comprising:

a callback status memory;

wherein the processor is operable to maintain a record of callback requests in the callback status memory, provide the callback request to the second client if the second client is logged onto the server, store the callback request in the record if the second client is logged off of the server, and provide the stored callback request to the second client if the second client subsequently logs onto the server.

25. The server of claim 23 wherein the processor is operable to receive a cancellation of the callback request from the second client.

26. The server of claim 23, further comprising:

a log-on status memory;

a callback status memory; and

wherein the processor is operable to store the identities of the first and second clients in the log-on status memory if the first and second clients, respectively, are logged onto the server, to obtain the log-on status of the second client from the log-on status memory in response to receiving the callback request from the first client, to provide the callback request to the second client if the second client is logged onto the server, and to store the callback request in the callback status memory if the second client is logged off of the server first .

27. A method for facilitating communication between a sending device and a receiving device, the method comprising:

allowing a message from the sending device to be directly routed past a server to the receiving device if the server is in first state; and

receiving the message from the sending device with the server and sending the received message from the server to the receiving device if the server is in the second state.

28. The method of claim 27, further comprising setting the state of the server in response to a message-initiation request from the sending device.

29. The method of claim 27, further comprising setting the server in the second state only if the sending device cannot send the message directly to the receiving device.

30. The method of claim 27, further comprising setting the state of the server in response to a message routing preference of a user.

31. A method, comprising directing a message from a sending device of a first type to a receiving device of a second type in response to a message routing preference of a user of the receiving device.

32. The method of claim 31, further comprising:

generating the user's message routing preference with a client running on a computer; and

providing the message routing preference to a server when the computer is logged on to the server, the server being in communication with the sending and receiving devices.

33. The method of claim 31 wherein the directing comprises directing the message to the receiving device upon occurrence of a condition specified by the message routing preference.

34. The method of claim 31 wherein the directing comprises directing the message to the receiving device when no client of the user is logged onto a server that is in communication with the sending and receiving devices.

35. The method of claim 31 wherein the directing comprises:

convert the message into a form that is compatible with the receiving device; and

sending the converted message to the receiving device.

36. The method of claim 31, further comprising:

allowing the sending device to send the message to a receiving device of the first type; and

wherein the directing comprises directing the message to the receiving device after the allowing and upon the occurrence of a condition specified by the message routing preference.

37. A method, comprising routing a message from a sending device to a first receiving device of a user in response to a message routing preference of the user, the sending device having directed the message to a second receiving device of the user.

38. The method of claim 37 wherein the routing comprises routing the message to the first receiving device upon occurrence of a condition specified by the message routing preference.

39. The method of claim 37 wherein:

the second receiving device comprises a desktop pager; and

the routing comprises routing the message to the first receiving device if no computer running the desktop pager is logged onto a server that is in communication with the sending and first receiving devices.

40. The method of claim 37 wherein:

the message is sent from the sending device in a form that is compatible with the second receiving device; and

the routing comprises converting the message into a form that is compatible with the first receiving device and sending the converted message to the first receiving device.

41. The method of claim 37 wherein:

the second device comprises a desktop pager;

the message routing preference specifies a period of user inactivity on a computer that is running the desktop pager; and

the routing comprises routing the message to the first receiving device after the period has elapsed.

42. A method, comprising storing on a server a log-on status for a client that has log-on privileges with respect to the server.

43. The method of claim 42 wherein the storing comprises storing a value in a memory location if the client is logged onto the server, the value corresponding to the client.

44. A method, comprising:

receiving a callback request from a first client that is logged onto the server; and

providing the callback request to a second client.

45. The method of claim 44, further comprising:

maintaining a record of callback requests;

providing the callback request to the second client if the second client is logged onto the server; and

storing the callback request in the record if the second client is logged off of the server

46. The method of claim 44, further comprising:

receiving a callback cancellation request from the second client; and

canceling the callback request in response to the cancellation request.

47. The method of claim 44, further comprising:

receiving a callback acknowledgement from the second client; and

notifying the first client that the second client is available.

48. The method of claim 44, further comprising:

maintaining a record of callback requests;

providing the callback request to the second client if the second client is logged onto the server;

storing the callback request in the record if the second client is logged off of the server; and

providing the stored callback request to the second client if the second client subsequently logs onto the server.

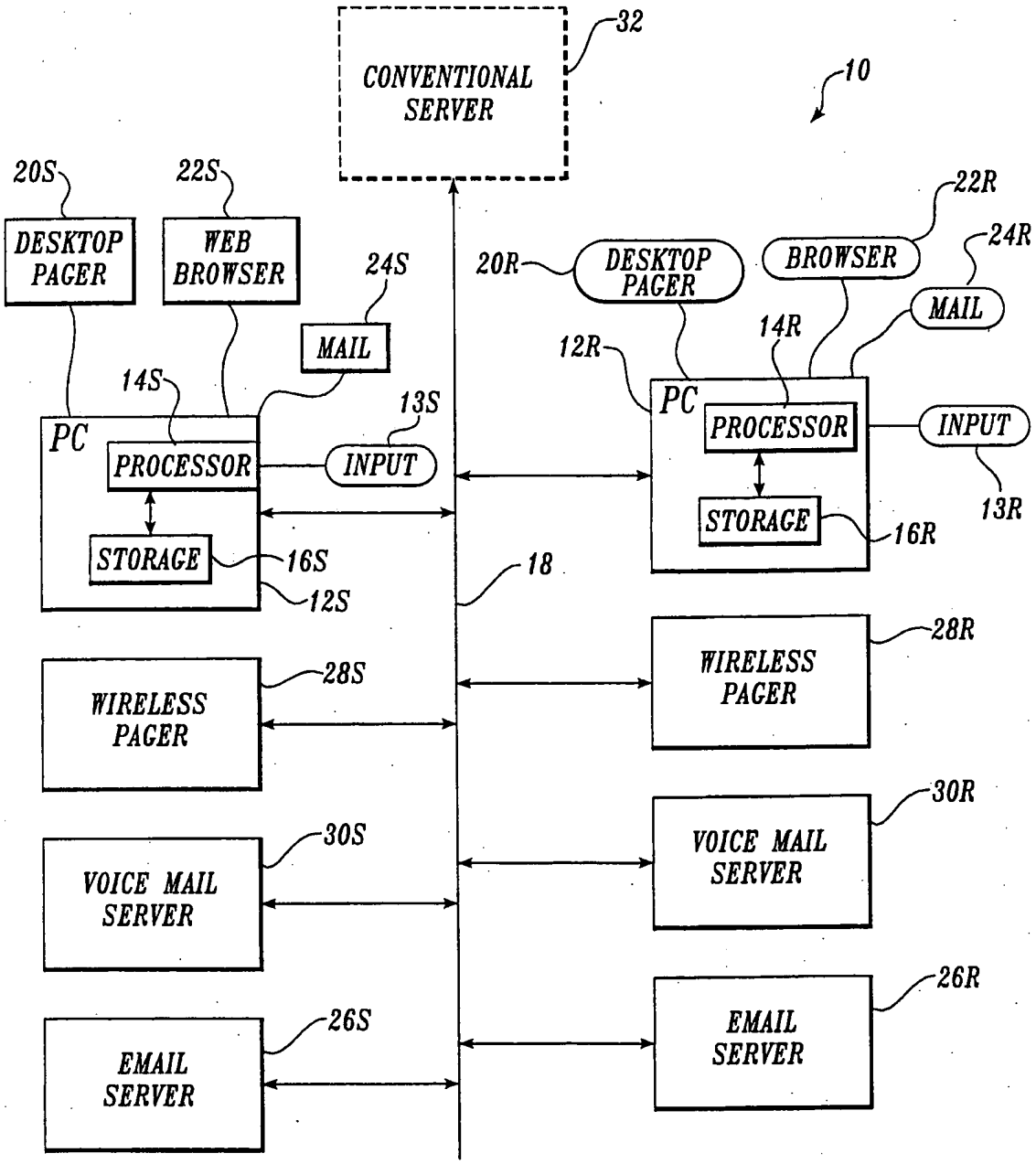


Fig. 1.
(PRIOR ART)

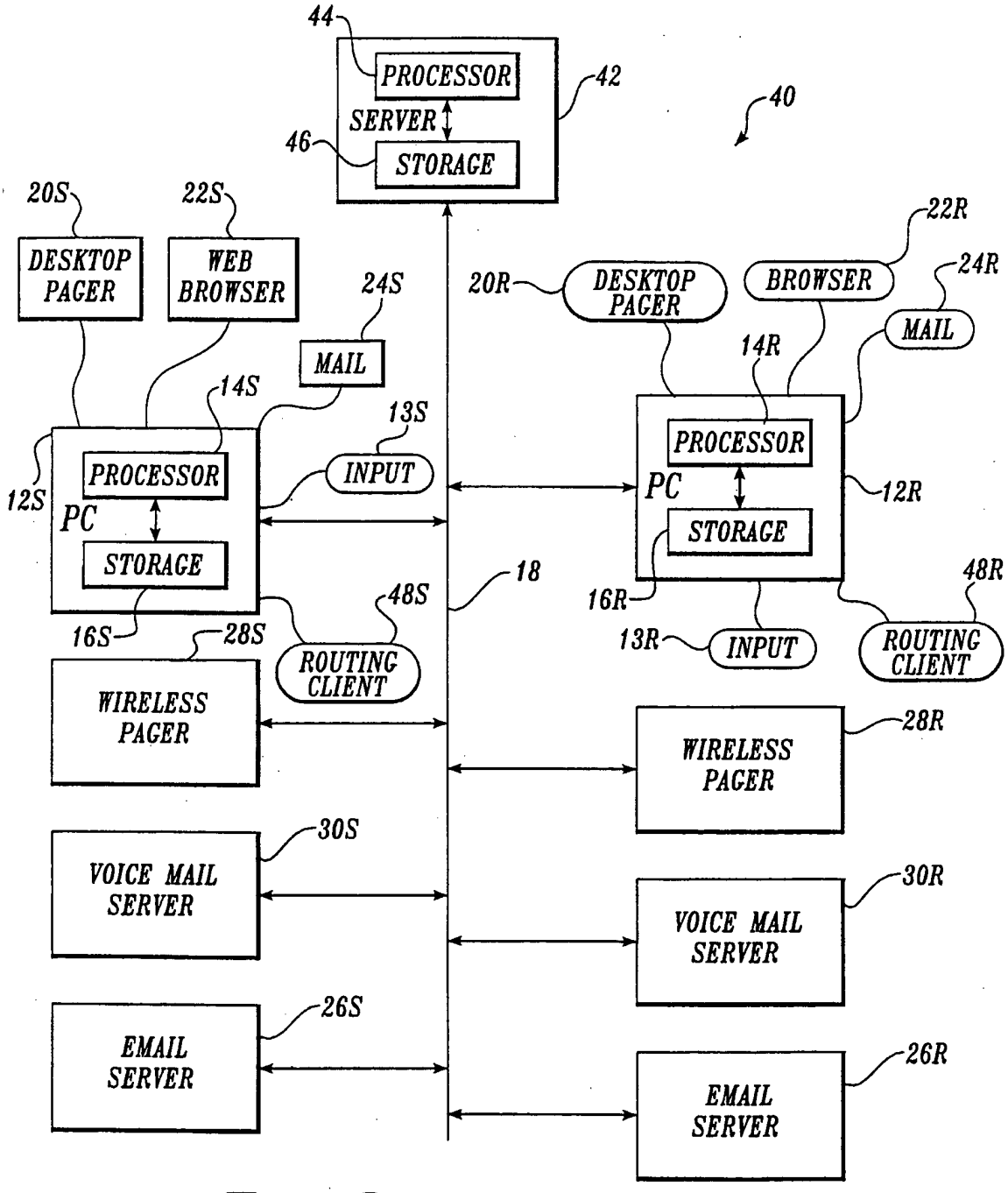

44
PROCESSOR
SERVER
STORAGE
46
42
40
20S
DESKTOP PAGER
22S
WEB BROWSER
24S
MAIL
14S
13S
INPUT
12S
PC
PROCESSOR
STORAGE
16S
48S
ROUTING CLIENT
28S
WIRELESS PAGER
18
30S
VOICE MAIL SERVER
26S
EMAIL SERVER
20R
DESKTOP PAGER
22R
BROWSER
24R
MAIL
14R
PROCESSOR
PC
STORAGE
12R
16R
13R
INPUT
48R
ROUTING CLIENT
28R
WIRELESS PAGER
30R
VOICE MAIL SERVER
26R
EMAIL SERVER

Fig. 2.

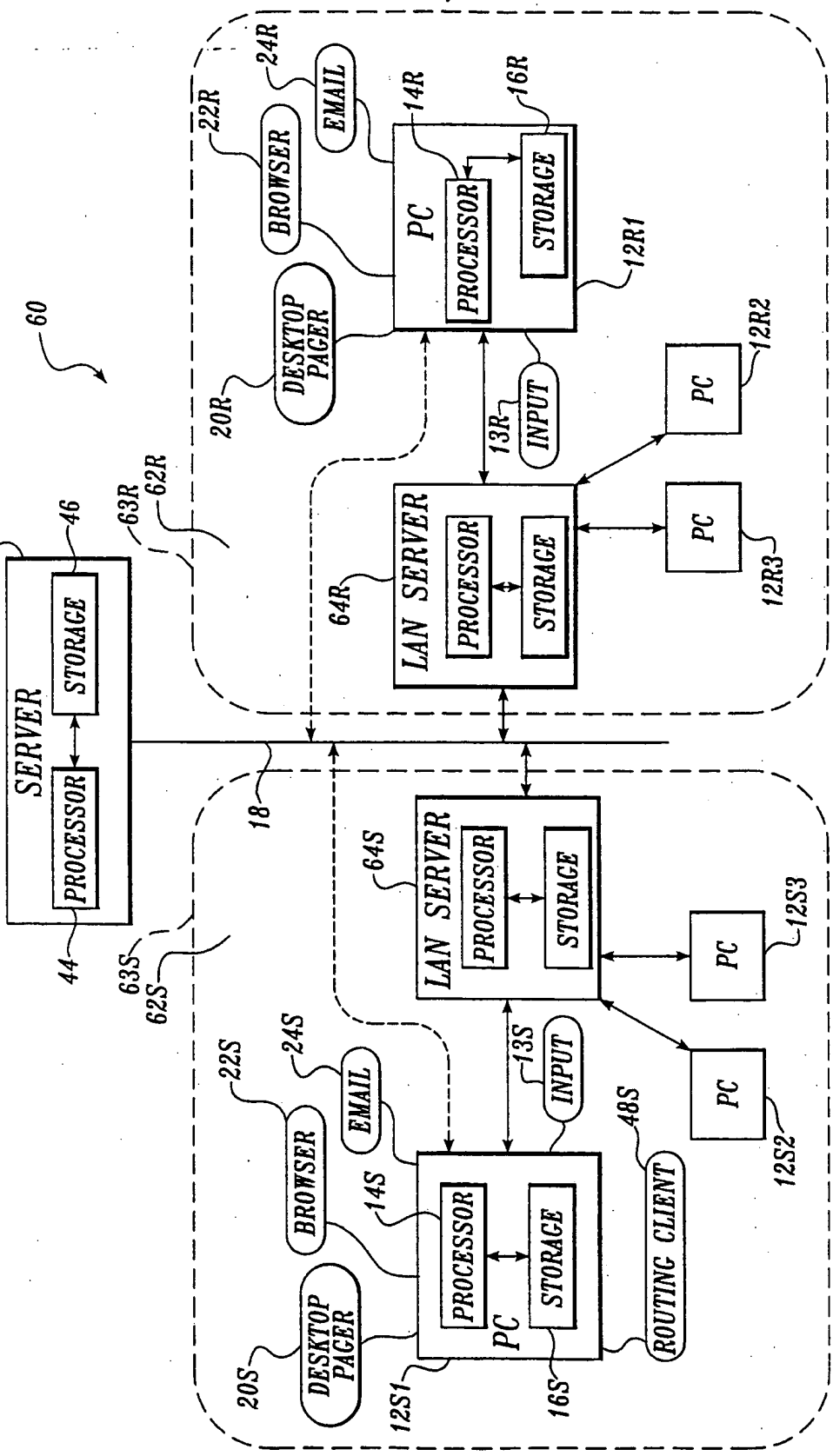

60
SERVER
PROCESSOR
STORAGE
42
44
46
18
63R
62R
63S
62S
LAN SERVER
PROCESSOR
STORAGE
64R
64S
PC
12R1
12R2
12R3
12S1
12S2
12S3
BROWSER
EMAIL
DESKTOP PAGER
22R
24R
20R
22S
24S
20S
INPUT
13R
13S
14R
14S
16R
16S
ROUTING CLIENT
48S

Fig. 3.

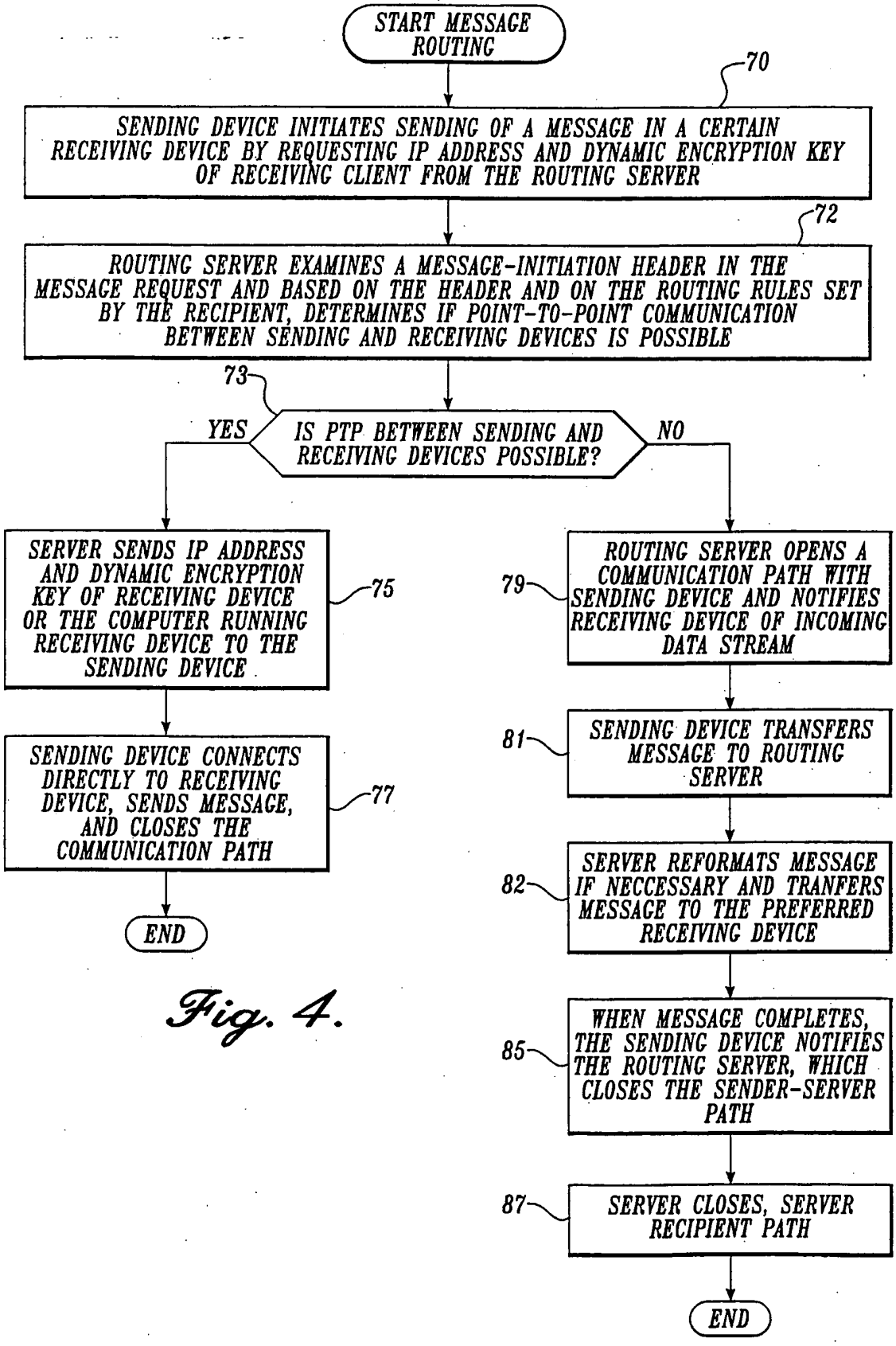

START MESSAGE ROUTING
70
SENDING DEVICE INITIATES SENDING OF A MESSAGE IN A CERTAIN RECEIVING DEVICE BY REQUESTING IP ADDRESS AND DYNAMIC ENCRYPTION KEY OF RECEIVING CLIENT FROM THE ROUTING SERVER
72
ROUTING SERVER EXAMINES A MESSAGE-INITIATION HEADER IN THE MESSAGE REQUEST AND BASED ON THE HEADER AND ON THE ROUTING RULES SET BY THE RECIPIENT, DETERMINES IF POINT-TO-POINT COMMUNICATION BETWEEN SENDING AND RECEIVING DEVICES IS POSSIBLE
73
IS PTP BETWEEN SENDING AND RECEIVING DEVICES POSSIBLE?
YES
NO
SERVER SENDS IP ADDRESS AND DYNAMIC ENCRYPTION KEY OF RECEIVING DEVICE OR THE COMPUTER RUNNING RECEIVING DEVICE TO THE SENDING DEVICE
75
SENDING DEVICE CONNECTS DIRECTLY TO RECEIVING DEVICE, SENDS MESSAGE, AND CLOSES THE COMMUNICATION PATH
77
END
79
ROUTING SERVER OPENS A COMMUNICATION PATH WITH SENDING DEVICE AND NOTIFIES RECEIVING DEVICE OF INCOMING DATA STREAM
81
SENDING DEVICE TRANSFERS MESSAGE TO ROUTING SERVER
82
SERVER REFORMATS MESSAGE IF NECCESSARY AND TRANFERS MESSAGE TO THE PREFERRED RECEIVING DEVICE
85
WHEN MESSAGE COMPLETES, THE SENDING DEVICE NOTIFIES THE ROUTING SERVER, WHICH CLOSES THE SENDER-SERVER PATH
87
SERVER CLOSES, SERVER RECIPIENT PATH
END

Fig. 4.

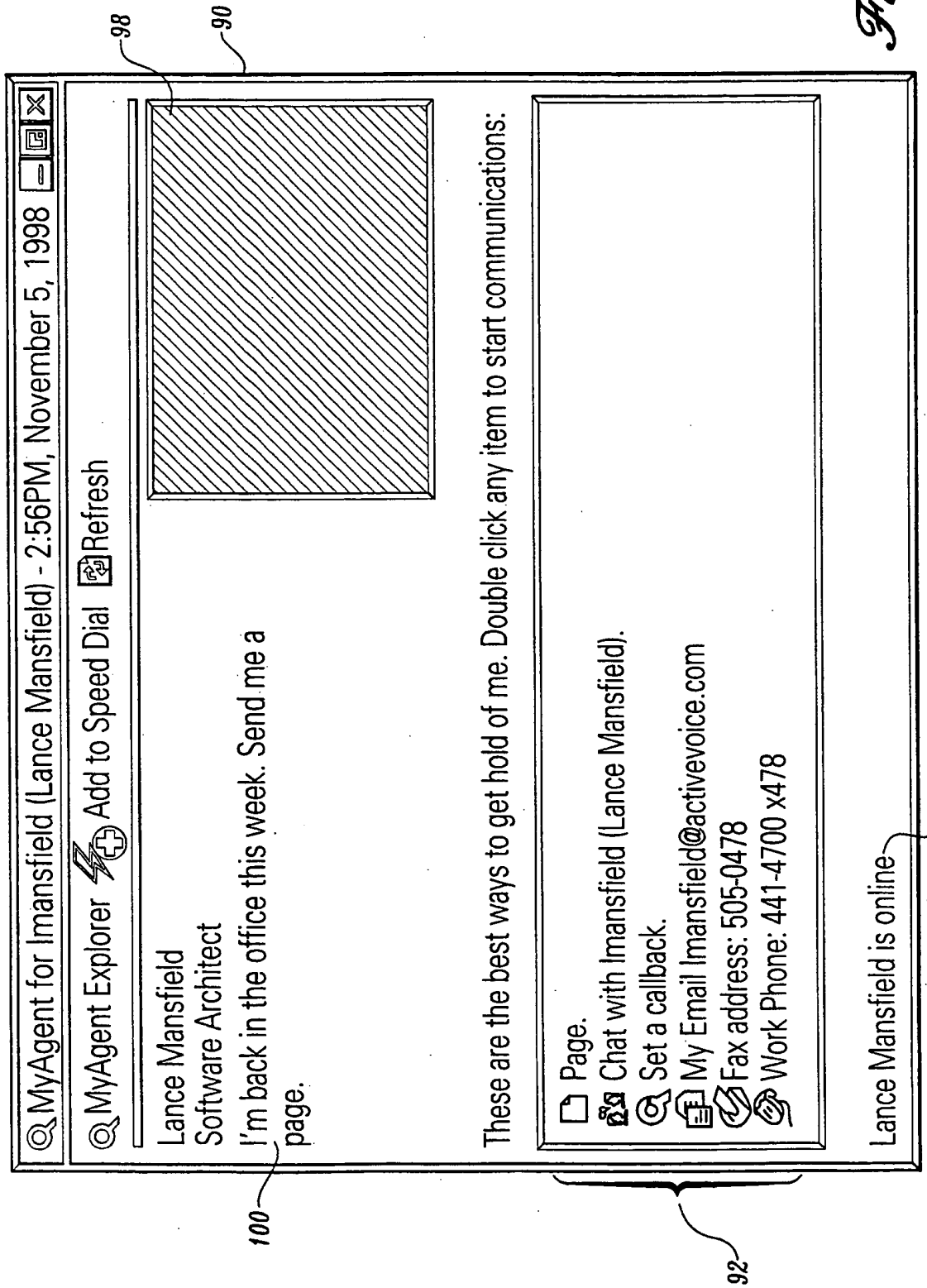

MyAgent for lmansfield (Lance Mansfield) - 2:56PM, November 5, 1998
MyAgent Explorer
Add to Speed Dial
Refresh
Lance Mansfield
Software Architect
I'm back in the office this week. Send me a page.
These are the best ways to get hold of me. Double click any item to start communications:
Page.
Chat with lmansfield (Lance Mansfield).
Set a callback.
My Email lmansfield@activevoice.com
Fax address: 505-0478
Work Phone: 441-4700 x478
Lance Mansfield is online
98
90
100
92
102

Fig. 5.

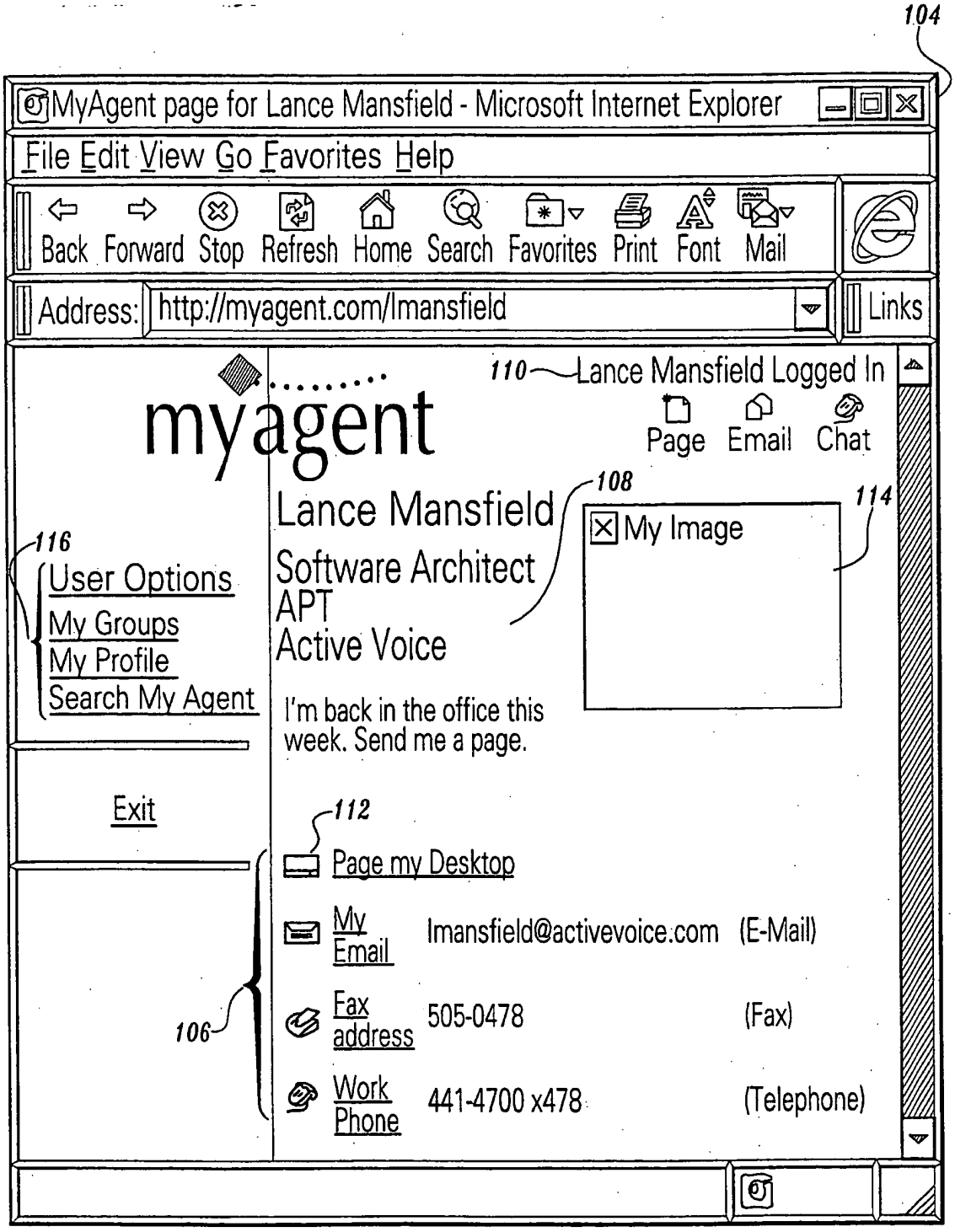

104
MyAgent page for Lance Mansfield - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Address: http://myagent.com/lmansfield
Links
myagent
110
Lance Mansfield Logged In
Page Email Chat
108
Lance Mansfield
Software Architect
APT
Active Voice
I'm back in the office this week. Send me a page.
114
My Image
116
User Options
My Groups
My Profile
Search My Agent
Exit
112
Page my Desktop
My Email lmansfield@activevoice.com (E-Mail)
Fax address 505-0478 (Fax)
Work Phone 441-4700 x478 (Telephone)
106

Fig. 6.

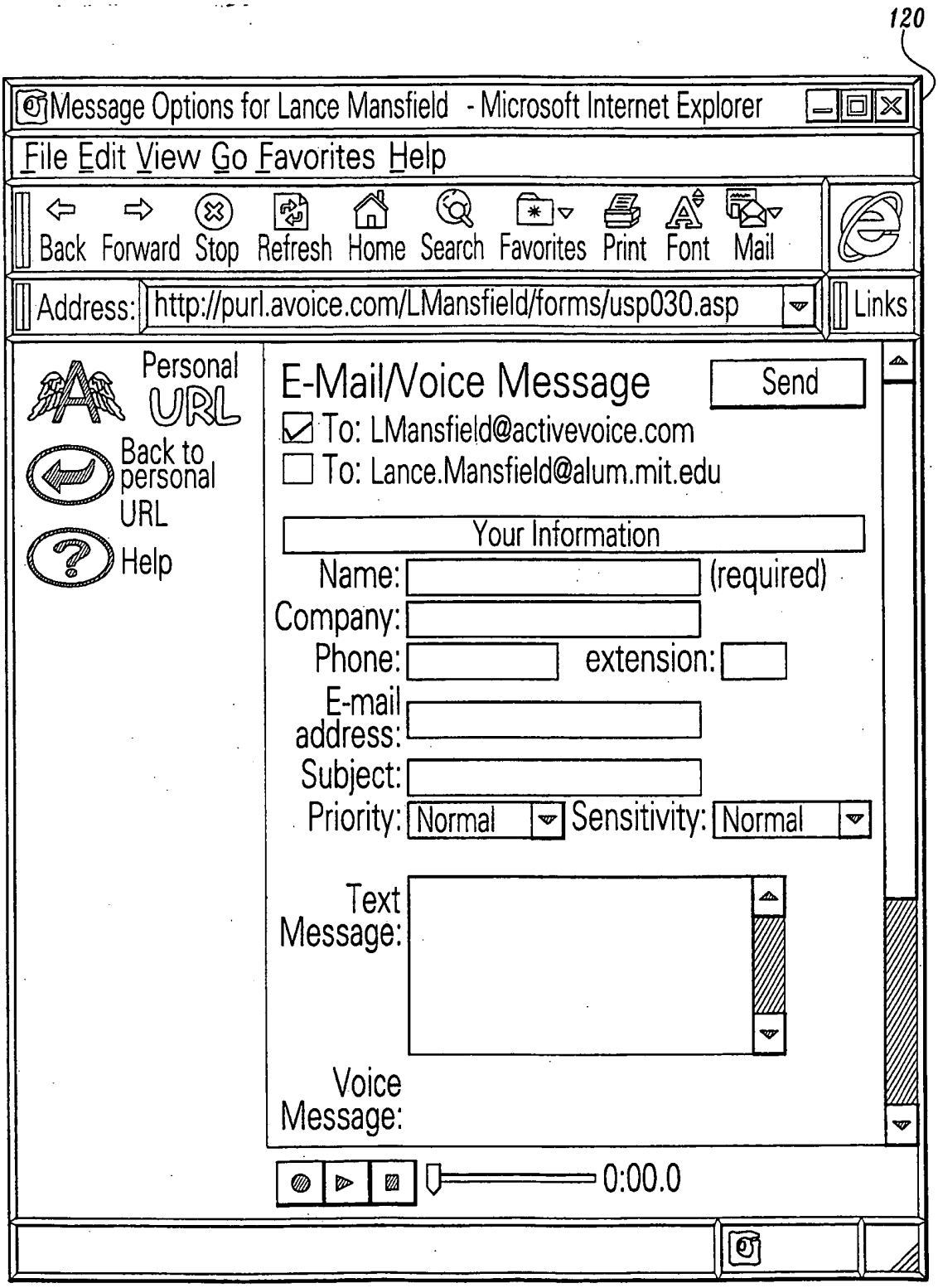

120
Message Options for Lance Mansfield - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Address: http://purl.avoice.com/LMansfield/forms/usp030.asp
Links
Personal URL
Back to personal URL
Help
E-Mail/Voice Message
Send
To: LMansfield@activevoice.com
To: Lance.Mansfield@alum.mit.edu
Your Information
Name: (required)
Company:
Phone: extension:
E-mail address:
Subject:
Priority: Normal Sensitivity: Normal
Text Message:
Voice Message:
0:00.0

Fig. 7.

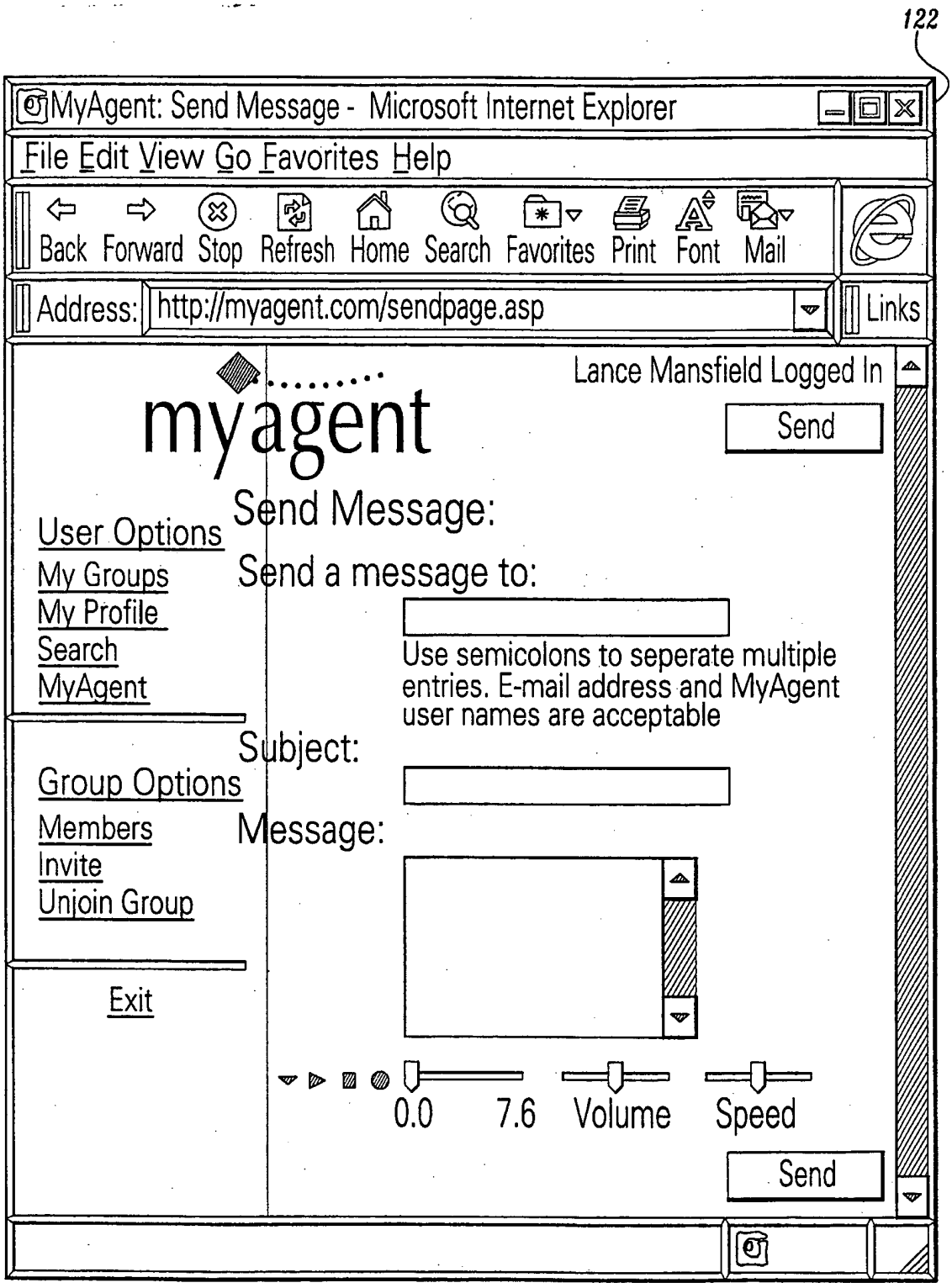

122
MyAgent: Send Message - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Address: http://myagent.com/sendpage.asp
Links
Lance Mansfield Logged In
myagent
Send
Send Message:
User Options
My Groups
My Profile
Search
MyAgent
Send a message to:
Use semicolons to seperate multiple entries. E-mail address and MyAgent user names are acceptable
Subject:
Group Options
Members
Invite
Unjoin Group
Message:
Exit
0.0
7.6
Volume
Speed
Send

Fig. 8.

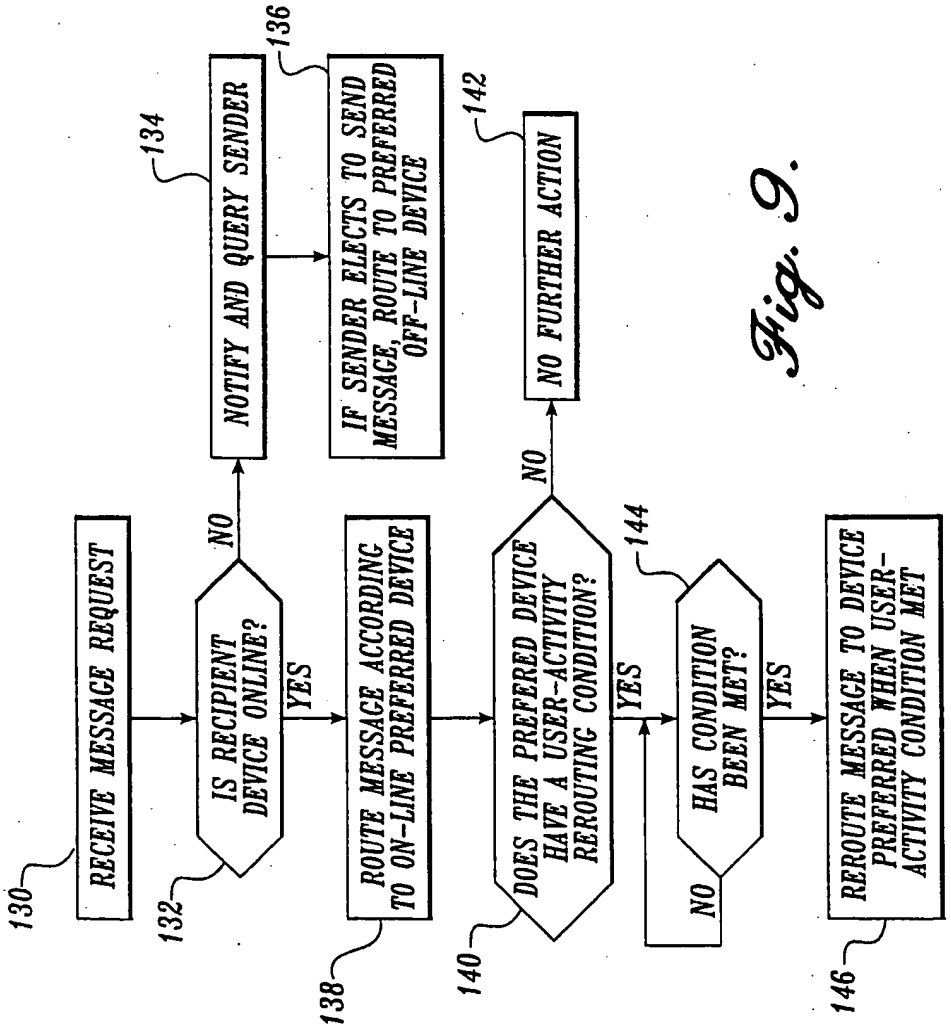

RECEIVE MESSAGE REQUEST
130
IS RECIPIENT DEVICE ONLINE?
132
NO
YES
NOTIFY AND QUERY SENDER
134
IF SENDER ELECTS TO SEND MESSAGE, ROUTE TO PREFERRED OFF-LINE DEVICE
136
ROUTE MESSAGE ACCORDING TO ON-LINE PREFERRED DEVICE
138
DOES THE PREFERED DEVICE HAVE A USER-ACTIVITY REROUTING CONDITION?
140
NO
NO FURTHER ACTION
142
YES
HAS CONDITION BEEN MET?
144
NO
YES
REROUTE MESSAGE TO DEVICE PREFERRED WHEN USER-ACTIVITY CONDITION MET
146

Fig. 9.

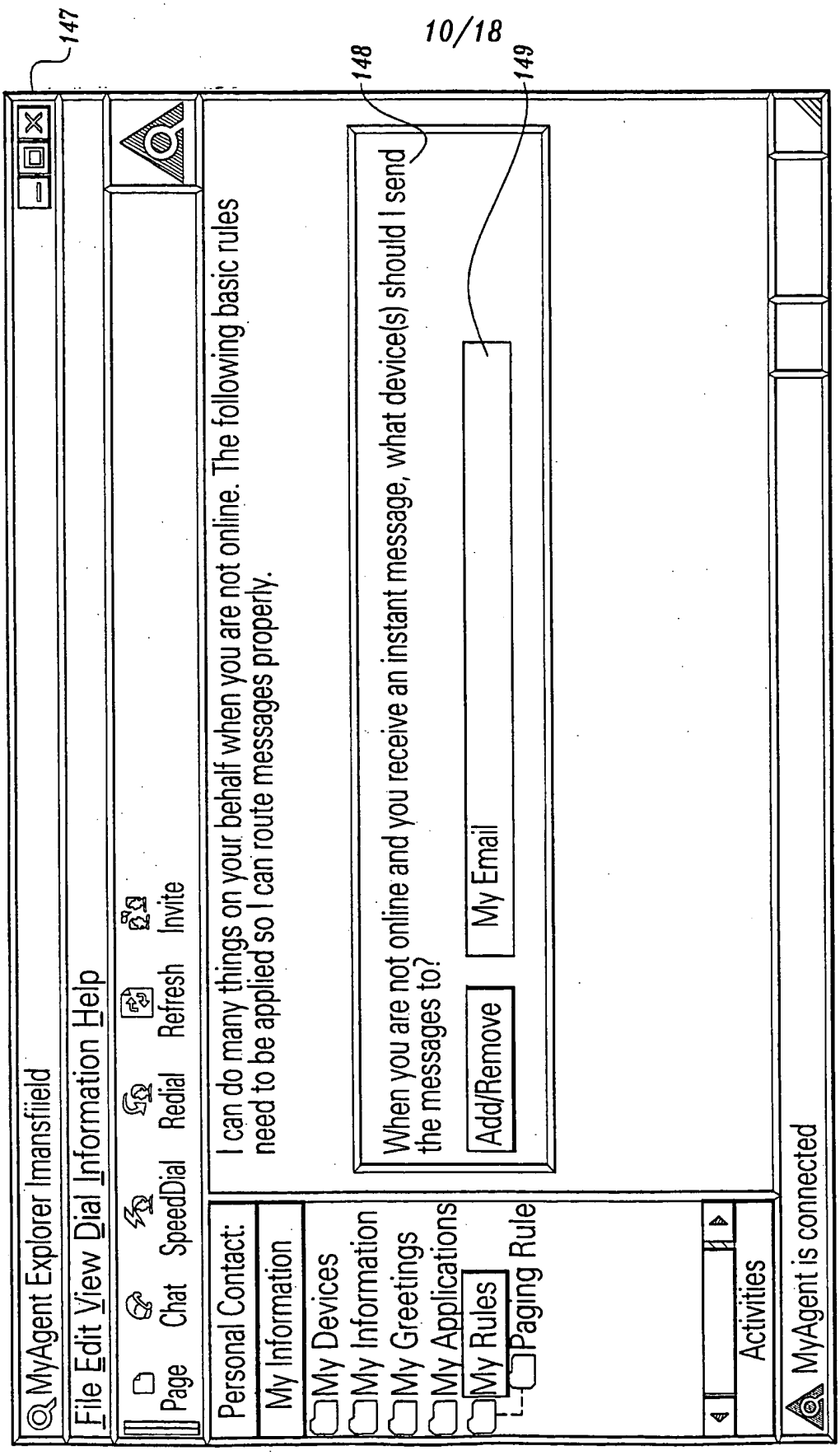

147
MyAgent Explorer Imansfiield
File Edit View Dial Information Help
Page
Chat
SpeedDial
Redial
Refresh
Invite
Personal Contact:
My Information
My Devices
My Information
My Greetings
My Applications
My Rules
Paging Rule
Activities
I can do many things on your behalf when you are not online. The following basic rules need to be applied so I can route messages properly.
148
When you are not online and you receive an instant message, what device(s) should I send the messages to?
Add/Remove
My Email
149
MyAgent is connected

Fig. 10.

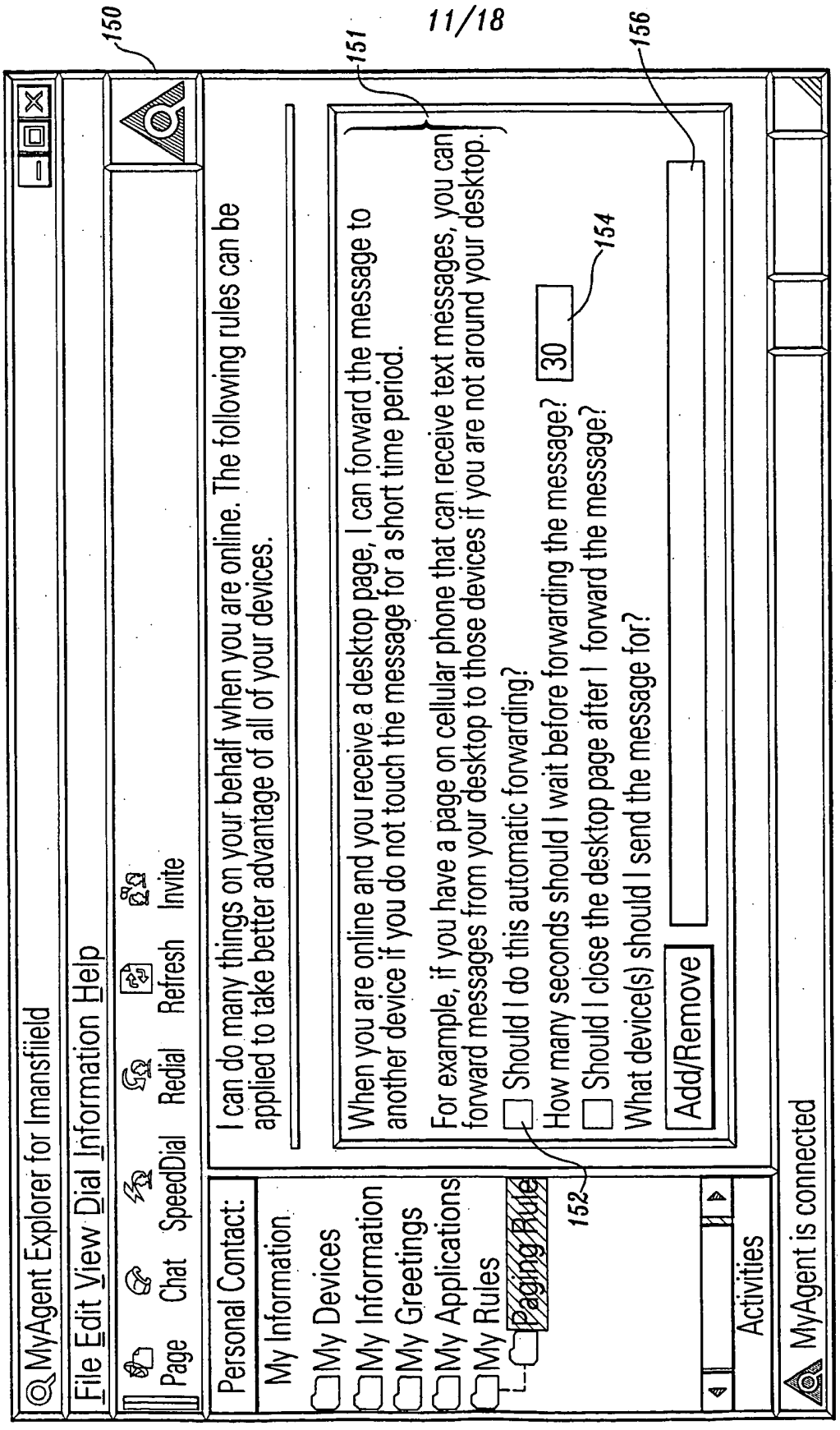

MyAgent Explorer for Imansfiield
File Edit View Dial Information Help
Page Chat SpeedDial Redial Refresh Invite
Personal Contact:
My Information
My Devices
My Information
My Greetings
My Applications
My Rules
Paging Rule
Activities
I can do many things on your behalf when you are online. The following rules can be applied to take better advantage of all of your devices.
When you are online and you receive a desktop page, I can forward the message to another device if you do not touch the message for a short time period.
For example, if you have a page on cellular phone that can receive text messages, you can forward messages from your desktop to those devices if you are not around your desktop.
Should I do this automatic forwarding?
How many seconds should I wait before forwarding the message? 30
Should I close the desktop page after I forward the message?
What device(s) should I send the message for?
Add/Remove
MyAgent is connected
150
151
152
154
156

Fig. 11.

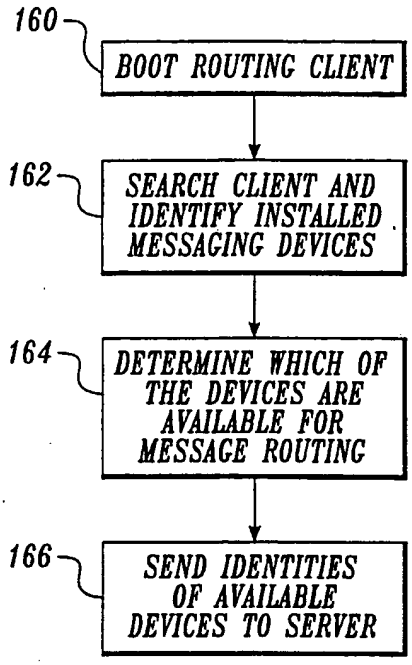

160
BOOT ROUTING CLIENT
162
SEARCH CLIENT AND IDENTIFY INSTALLED MESSAGING DEVICES
164
DETERMINE WHICH OF THE DEVICES ARE AVAILABLE FOR MESSAGE ROUTING
166
SEND IDENTITIES OF AVAILABLE DEVICES TO SERVER

Fig. 12.

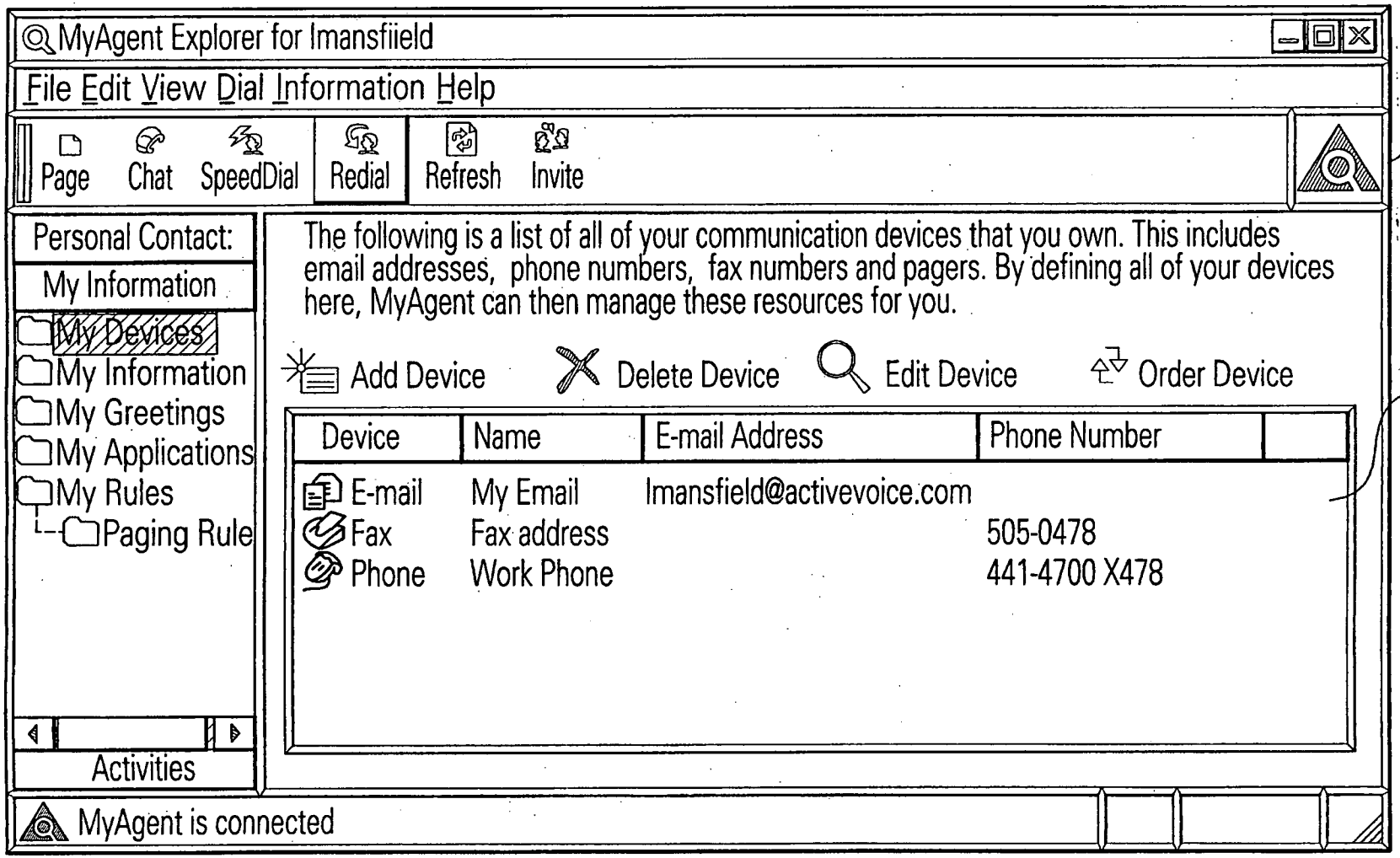

MyAgent Explorer for lmansfiield
File Edit View Dial Information Help
Page
Chat
SpeedDial
Redial
Refresh
Invite
170
Personal Contact:
My Information
My Devices
My Information
My Greetings
My Applications
My Rules
Paging Rule
Activities
The following is a list of all of your communication devices that you own. This includes email addresses, phone numbers, fax numbers and pagers. By defining all of your devices here, MyAgent can then manage these resources for you.
Add Device
Delete Device
Edit Device
Order Device
172
Device
Name
E-mail Address
Phone Number
E-mail
My Email
lmansfield@activevoice.com
Fax
Fax address
505-0478
Phone
Work Phone
441-4700 X478
MyAgent is connected

Fig. 13.

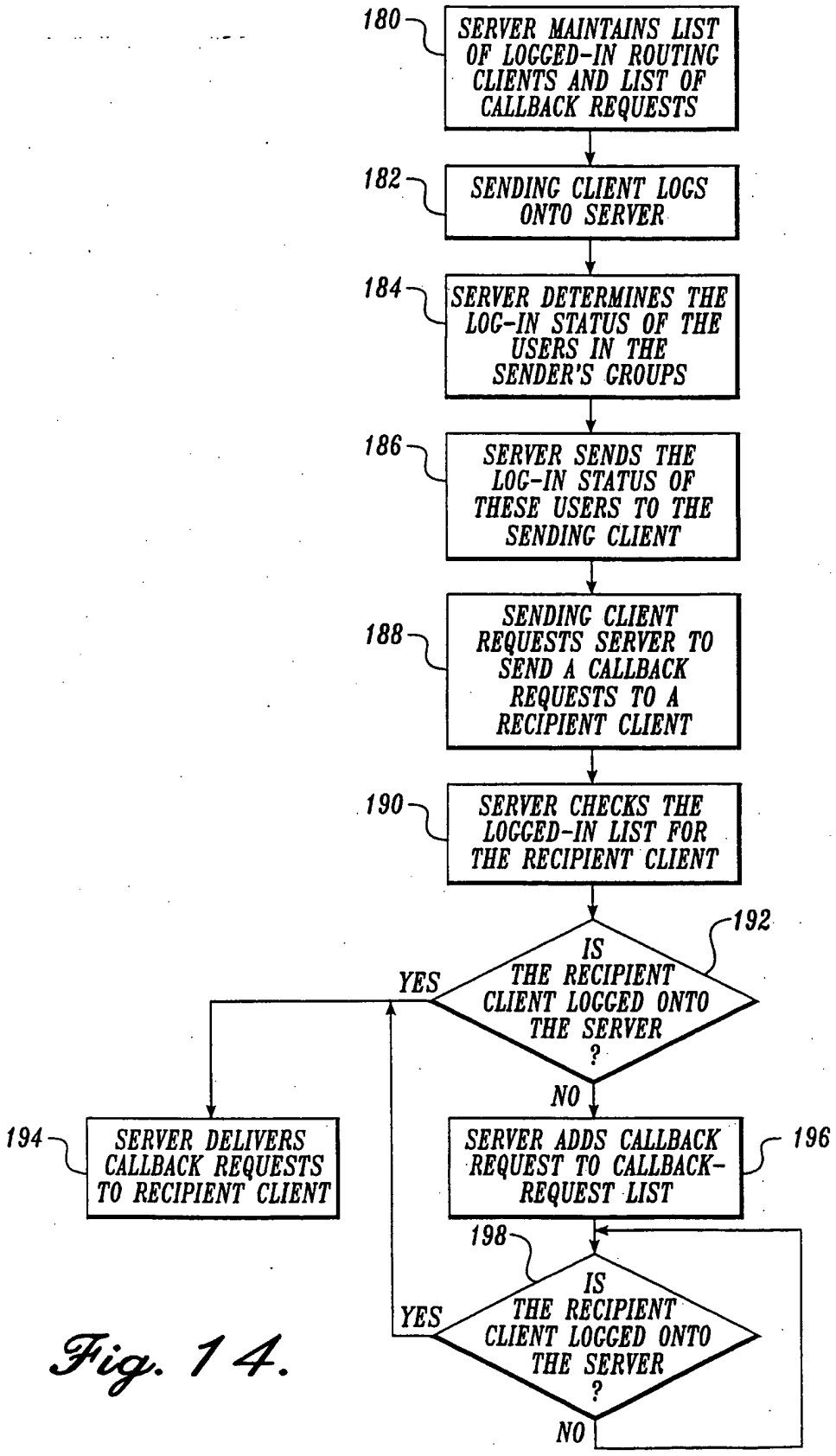

180
SERVER MAINTAINS LIST OF LOGGED-IN ROUTING CLIENTS AND LIST OF CALLBACK REQUESTS
182
SENDING CLIENT LOGS ONTO SERVER
184
SERVER DETERMINES THE LOG-IN STATUS OF THE USERS IN THE SENDER'S GROUPS
186
SERVER SENDS THE LOG-IN STATUS OF THESE USERS TO THE SENDING CLIENT
188
SENDING CLIENT REQUESTS SERVER TO SEND A CALLBACK REQUESTS TO A RECIPIENT CLIENT
190
SERVER CHECKS THE LOGGED-IN LIST FOR THE RECIPIENT CLIENT
192
IS THE RECIPIENT CLIENT LOGGED ONTO THE SERVER ?
YES
NO
194
SERVER DELIVERS CALLBACK REQUESTS TO RECIPIENT CLIENT
196
SERVER ADDS CALLBACK REQUEST TO CALLBACK-REQUEST LIST
198
IS THE RECIPIENT CLIENT LOGGED ONTO THE SERVER ?
YES
NO

Fig. 14.

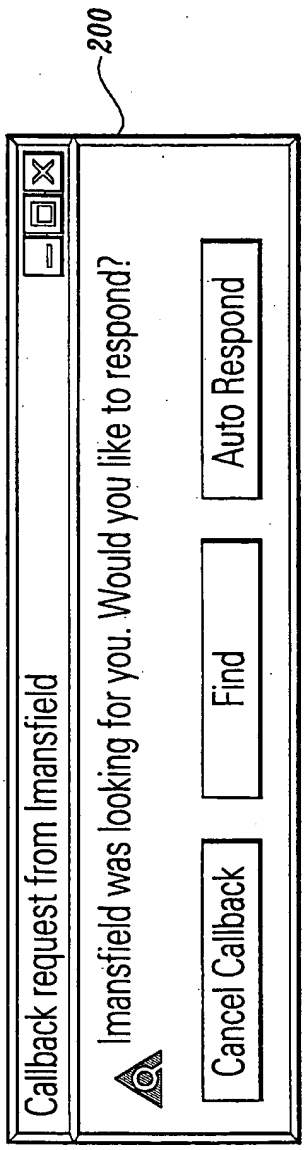

200
Callback request from Imansfield
Imansfield was looking for you. Would you like to respond?
Cancel Callback
Find
Auto Respond

Fig. 15.

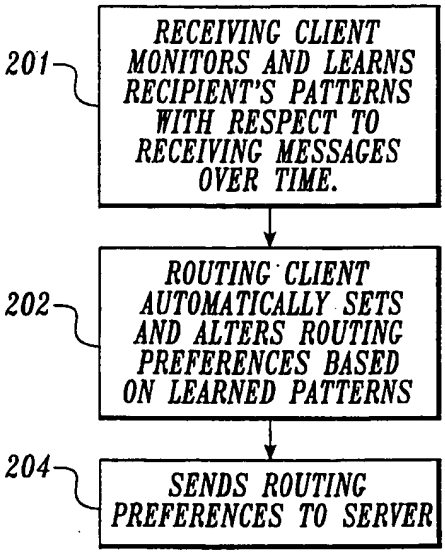

201
RECEIVING CLIENT MONITORS AND LEARNS RECIPIENT'S PATTERNS WITH RESPECT TO RECEIVING MESSAGES OVER TIME.
202
ROUTING CLIENT AUTOMATICALLY SETS AND ALTERS ROUTING PREFERENCES BASED ON LEARNED PATTERNS
204
SENDS ROUTING PREFERENCES TO SERVER

Fig. 16.

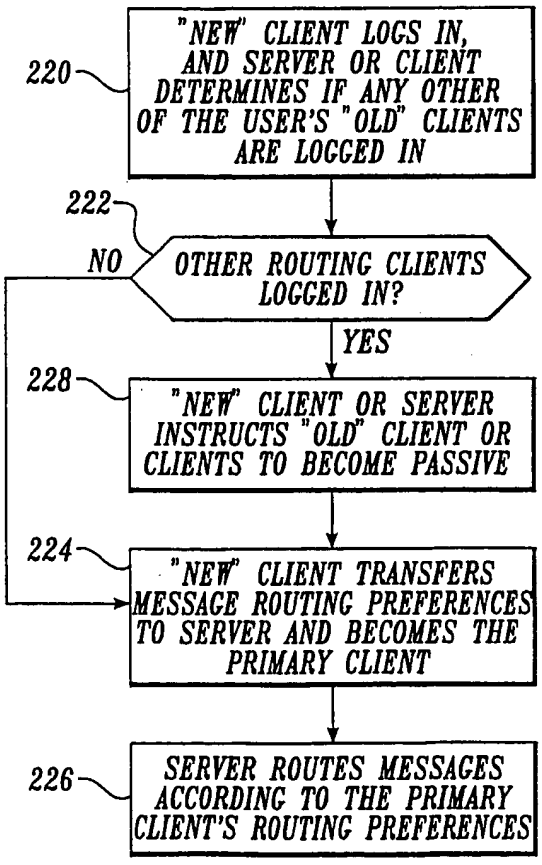

220
"NEW" CLIENT LOGS IN, AND SERVER OR CLIENT DETERMINES IF ANY OTHER OF THE USER'S "OLD" CLIENTS ARE LOGGED IN
222
NO
OTHER ROUTING CLIENTS LOGGED IN?
YES
228
"NEW" CLIENT OR SERVER INSTRUCTS "OLD" CLIENT OR CLIENTS TO BECOME PASSIVE
224
"NEW" CLIENT TRANSFERS MESSAGE ROUTING PREFERENCES TO SERVER AND BECOMES THE PRIMARY CLIENT
226
SERVER ROUTES MESSAGES ACCORDING TO THE PRIMARY CLIENT'S ROUTING PREFERENCES

Fig. 18.

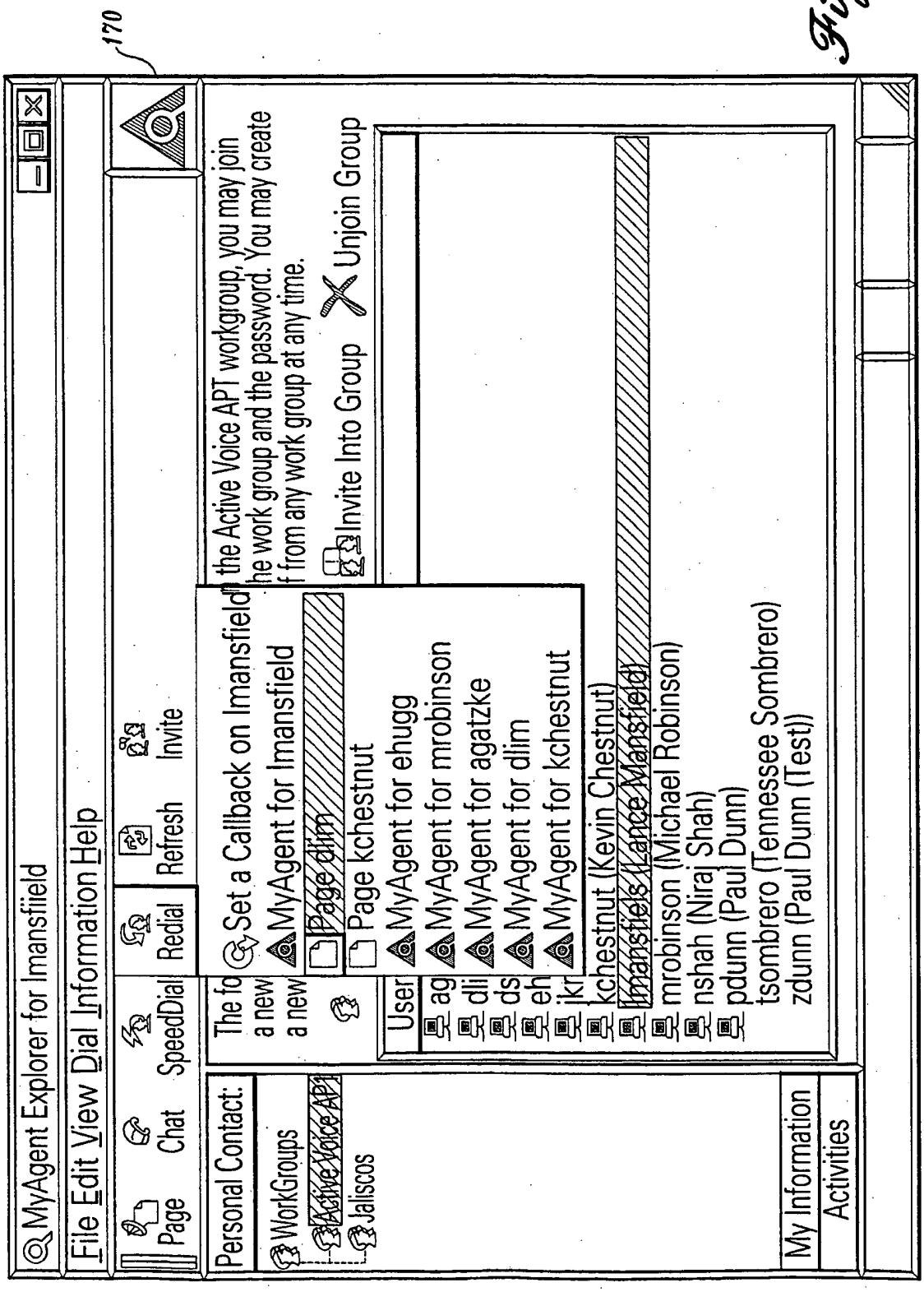

170
MyAgent Explorer for Imansfiield
File Edit View Dial Information Help
Page
Chat
SpeedDial
Redial
Refresh
Invite
Personal Contact:
WorkGroups
Active Voice APT
Jaliscos
The fo
a new
a new
Set a Callback on Imansfield
MyAgent for Imansfield
Page dim
Page kchestnut
User
ag
dli
ds
eh
ikr
MyAgent for ehugg
MyAgent for mrobinson
MyAgent for agatzke
MyAgent for dlim
MyAgent for kchestnut
kchestnut (Kevin Chestnut)
Imansfiels (Lance Mansfield)
mrobinson (Michael Robinson)
nshah (Niraj Shah)
pdunn (Paul Dunn)
tsombrero (Tennessee Sombrero)
zdunn (Paul Dunn (Test))
the Active Voice APT workgroup, you may join
he work group and the password. You may create
f from any work group at any time.
Invite Into Group
Unjoin Group
My Information
Activities

Fig. 17.

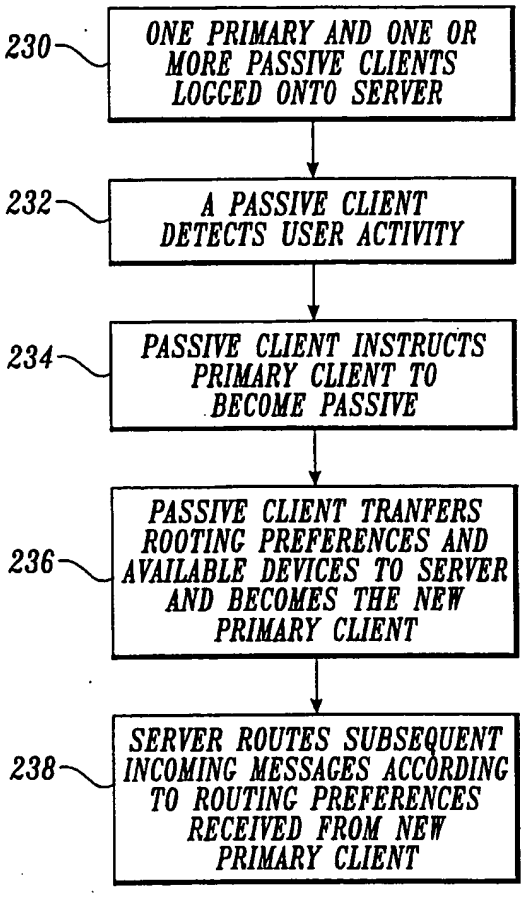

230
ONE PRIMARY AND ONE OR MORE PASSIVE CLIENTS LOGGED ONTO SERVER
232
A PASSIVE CLIENT DETECTS USER ACTIVITY
234
PASSIVE CLIENT INSTRUCTS PRIMARY CLIENT TO BECOME PASSIVE
236
PASSIVE CLIENT TRANFERS ROOTING PREFERENCES AND AVAILABLE DEVICES TO SERVER AND BECOMES THE NEW PRIMARY CLIENT
238
SERVER ROUTES SUBSEQUENT INCOMING MESSAGES ACCORDING TO ROUTING PREFERENCES RECEIVED FROM NEW PRIMARY CLIENT

Fig. 19.